



Курсова работа

На тема:

doElections – система за онлайн избори

За курса:

Софтуерни архитектури и разработка на софтуер,

Катедра „Софтуерни технологии“

Изготвено от:

Калоян Николов, 62252

Светлин Попиванов, 62275

Андрей Стоев, 62369

София, май 2020 г.

Съдържание	2
1. Въведение	4
1.1. Обща информация за текущия документ	4
1.1.1. Предназначение	4
1.1.2. Описание на използваните структури на архитектурата	4
1.1.2.1. Употреба на модули	4
1.1.2.2. Структура на процесите	4
1.1.2.3. Внедряване	5
1.1.3. Структура на документа	5
1.2. Общи сведения за системата	5
1.2.1. Същност на системата	5
1.2.2. Целева група	6
1.2.3. Качествени изисквания	6
2. Декомпозиция на модули	7
а) общ вид на декомпозицията на модули на системата	7
б) контекстна диаграма	7
с) Подробно описание на всеки модул	7
1. Потребител	8
2. Сървър	15
3. Техническа поддръжка	23
4. Мениджър на база данни	26
3. Описание на допълнителни структури	27
3.1. Структура на употреба на модули	27
3.1.1. Първично представяне	27
3.1.2. Описание на елементите и връзките	28
3.1.2.1. Връзки между основните модули в системата	28
3.1.2.1.1. Връзка между Потребител и Сървър	28
3.1.2.1.2. Връзка между Сървър и Мениджър на база данни	29
3.1.2.1.3. Връзка между Сървър и Техническа поддръжка	29
3.1.2.1.4. Връзка между Техническа поддръжка и Мениджър на база данни	29
3.1.2.2. Връзки в основните модули на системата	29
3.1.2.2.1. Потребител	30
3.1.2.2.2. Сървър	32
3.1.2.2.3. Техническа поддръжка	35
3.1.2.2.4. Мениджър на база данни	37
3.1.3. Описание на обкръжението	38
3.1.3.1. Връзка на системата с ЕСГРАОН	38
3.1.3.2. Връзка на системата с GMAIL	38
3.2. Структура на процесите	39
3.2.1. Първично представяне	39
3.2.2. Описание на елементите и връзките	40
3.2.3. Описание на обкръжението	41
3.2.4. Описание на възможните вариации	41
3.3. Внедряване	42
3.3.1. Първично представяне	42

3.3.2. Описание на елементите и връзките	42
3.3.3. Описание на обкръжението	44
3.3.4. Описание на възможните вариации	44
4. Архитектурна обосновка	45
4.1. Видове потребители	45
4.2. Начини за гласуване	45
4.3. Данни от външни системи	46
4.4. Необвързване на самоличността на гласоподавателя с неговия вот.....	46
4.5. Наличност.....	47
4.6. Интегритет на данните.....	48
4.7. Натоварване и производителност.....	48

1. Въведение

1.1. Обща информация за текущия документ

1.1.1. Предназначение

Настоящият документ цели да запознае читателят със системата за електронно гласуване doElections, нейните основните драйвери и архитектура, както и взетите архитектурни решения, които позволяват осигуряването на нужните качествени изисквания. Подробното описание на софтуерната архитектура чрез различни структури е полезно за всички заинтересовани лица - както разработчиците на системата, така и за клиентите ѝ, които ще следят начина на реализацията на системата. Може да послужи за шаблон, чрез който главният архитект да защити качествените характеристики на системата пред заинтересованите лица.

1.1.2. Описание на използваните структури на архитектурата.

1.1.2.1. Употреба на модули

Структурата на употреба на модули е включена в този документ, поради силното взаимодействие между модулите, изграждащи системата doElections. Тази структура е важна част от архитектурата на системата, защото описва детайлно статичните връзки между отделните модули и подмодули, изграждащи системата. По този начин, тя се явява естествено допълнение на също представената в документа декомпозиция на модули. Тази структура ще бъде изключително полезна за разработчиците и тестерите при добавянето и последвалото изпитване на дадена функционалност, тъй като се показва кои модули ще бъдат засегнати при евентуални промени по системата.

1.1.2.2. Структура на процесите

Едни от основните изисквания на системата са осигуряването на сигурността на преноса на данни за гласовете на гласоподавателите и наличността на системата колкото се може повече по време на изборния ден. Структурата на процесите ще ни помогне за онагледяване на това какви данни се изпращат между отделните модули и как те се изпращат - ако те не се изпращат по достатъчно добър начин, то това ще индикира, че е необходимо задължително подобрение в комуникацията в системата doElections. Също така, ако повечето заявки в системата преминават през един модул, то тази структура ще покаже и това натоварване, което ще помогне на разработчиците да определят къде би могло да скалират системата. Структурата на процесите би била полезна и за определянето на производителността на системата, защото чрез нея разработчиците биха се ориентирали къде системата се нуждае от подобрение в комуникацията или в използването на някакъв алгоритъм. Тази структура ще служи за следните заинтересовани лица: разработчиците, QA специалистите, клиентите, анализаторите и инвеститорите.

1.1.2.3. Внедряване

Тази структура ще представи връзката на софтуера с хардуера как той се разпределя върху хардуерните компоненти, както и какви са връзките с външните системи. От декомпозицията на модулите всеки отделен модул е разпределен на различно устройство, като някои от модулите са репликирани на няколко хардуерни устройства. Тъй като това онагледяване би отежнило твърде много структурата на модулите, то ще е най-подходящо да се представи именно тук. Тази структура ще допълни добре вече представените структури за декомпозиция на модули и употреба на модули. Структурата на внедряването ще бъде полезна за разкриването и начина на постигане на някои от ключовите качествени изисквания свързани със системата и желаното ѝ функциониране.

1.1.3. Структура на документа

Настоящият документ е разделен на 4 секции:

Секция 1: Това е въвеждащата секция, която включва обща информация - съдържа общи сведения за документа и системата и също така е представена мотивацията за всяка една от включените в следващите секции структури.

Секция 2: Съдържа подробно описание на структурата Декомпозиция на модули.

Секция 3: Съдържа допълнителните структури, които ще са от полза за читателите на документа: употреба на модули, структура на процесите и внедряване.

Секция 4: Съдържа архитектурна обосновка - описани са драйверите на системата и решенията, които са взети с цел да бъдат постигнати.

1.2. Общи сведения за системата

1.2.1. Същност на системата

Системата doElections дава възможност за осъществяване на електронни гласувания с най-различни мащаби - от средни и големи организации до избори в цели държави или дори групи от държави.

Системата предлага надежден вариант за осъществяване на правителствени избори и референдуми в ситуации, когато наличните алтернативи за гласуване не се приемат с достатъчно доверие от избирателите. Новата алтернатива, представена от системата, ще може да бъде предоставена на по-ниска цена от съществуващите начини за гласуване. Тя осигурява по-добра гъвкавост и по-висока скорост на изборния процес. Също така е важно да отбележим, че този вид гласуване допълва и подобрява текущите начини за осъществяване на избори, без да ги отхвърля.

Системата е лесна за използване като има предвидени функционалности за

добавяне на административни лица и кандидати, които да бъдат избирани. Това ще забърза подготовката и ще улесни провеждането на изборите.

1.2.2. Целева група

Системата е предназначена за осъществяването на избори, в които участниците са от различни градове, региони или дори страни. Гласуващите могат да имат най-разнообразни професии и степени на образование. Също така системата е изключително полезна за тези, които са на работа в изборния ден или изпитват трудност да отидат до избирателна секция - doElections им предоставя лесен и бърз начин за даване на своя вот.

Системата е предназначена за държави и организации, които са готови да преобразят стандартното гласуване чрез хартиени бюлетини в един нов иновативен онлайн метод, който позволява както гласуване от вкъщи чрез приложение на компютър/лаптоп, така и на място чрез употребата на терминали.

1.2.3. Качествени изисквания

1.2.3.1. Системата трябва да бъде 100% налична в рамките на даден изборен ден. Обикновено това е период от около 14-15 часа. В рамките на това време трябва да се толерират всички откази в хардуер, софтуер, комуникационно оборудване и т.н.

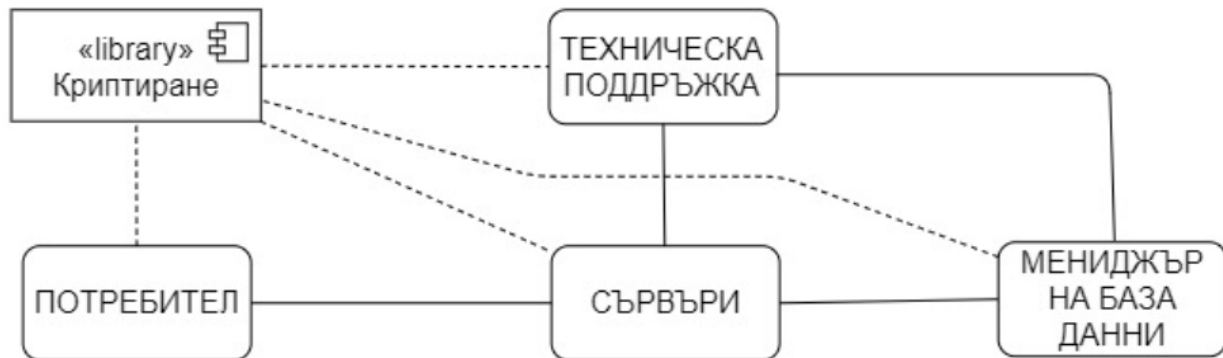
1.2.3.2. Системата трябва да гарантира 100% интегритет на данните по време на изборния ден.

1.2.3.3. Системата трябва да издържа натоварване от 100 000 едновременни заявки.

1.2.3.4. При генериране на заявка към системата без значение от натоварването, системата трябва да връща отговор в рамките на 1 секунда.

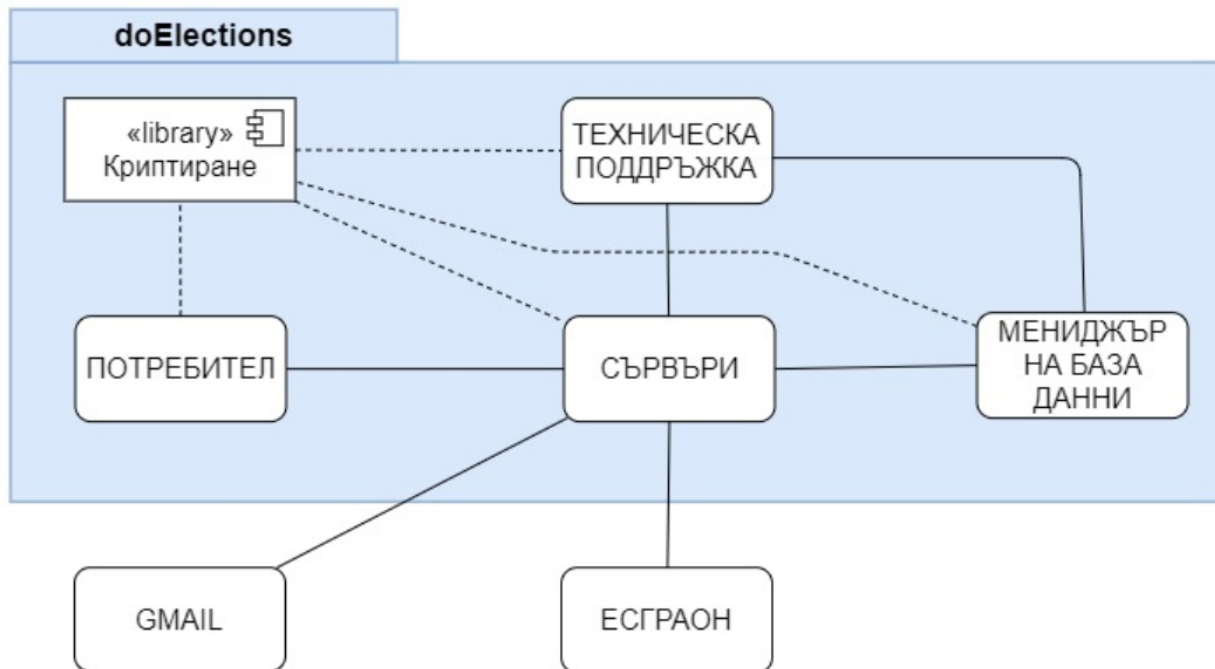
2.Декомпозиция на модули

а) общ вид на декомпозицията на модули на системата



Всеки модул от системата doElections е нейно логическо ядро и всеки от тях е декомпозиран до подмодули и класове, като подмодулите също са декомпозирани до класове.

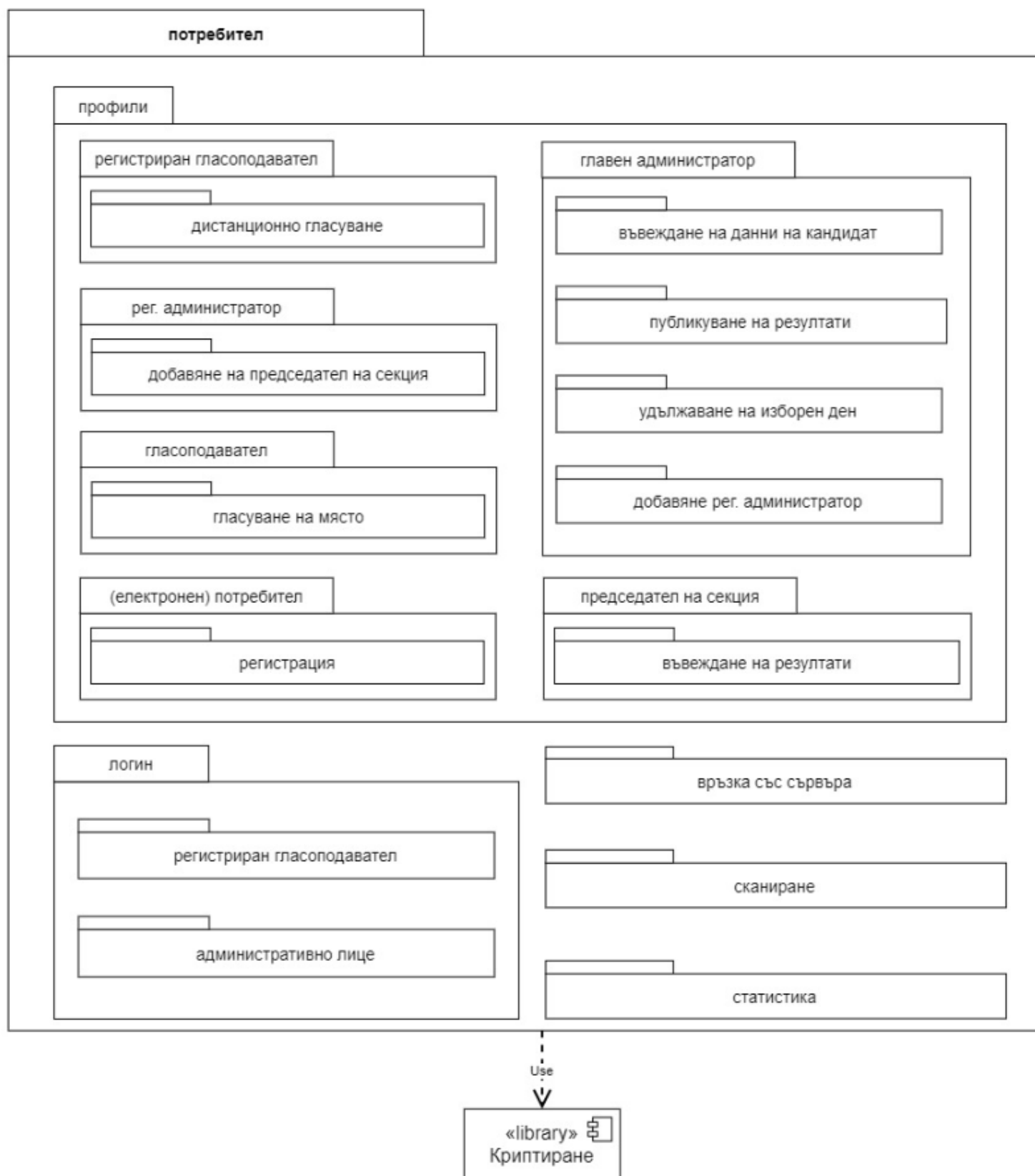
б) контекстна диаграма



с) Подробно описание на всеки модул

В следващите описания на интерфейсите ще подразбираме, че всяка една заявка, която се изпраща от текущия модул към друг ще бъде криптирана чрез вградената в него библиотека “криптиране” и ще бъде декриптирана в модула получател чрез същата тази библиотека.

1. Потребител



- **предназначение на модула** - този модул контролира интеракцията между системата и нейните потребители посредством съответните интерфейси на функционалности, които се предоставят чрез сървърите на doElections.
- **основни отговорности на модула:**
 - осигурява добро потребителско изживяване
 - осигурява сигурна и качествена връзка със сървър на doElections
 - приемането на глас от регистриран гласоподавател се осъществява атомарно.

Описание на интерфейсите на модула:

- *профили*
 - *регистриран гласоподавател*
 - *дистанционно гласуване*
 - *Vote(choice: int, KEP: String)*
Входни данни: код, показващ за кого е гласът, KEP
Изходни данни: код, показващ за кого е гласът и заявка за маркиране на регистрирания гласоподавател като гласувал
Ограничения при употреба: всеки гласоподавател може да гласува само веднъж в рамките на изборния ден
Грешки и изключения: при невалиден подаден KEP или липса на право на глас, системата ще връща съобщение за грешка
Зависимост от други елементи: "връзка със сървъра"
Пример за употреба: Регистриран гласоподавател избира опцията да гласува и удостоверява своята самоличност посредством KEP. Изпраща се заявка към "връзка със сървъра" с цел да се провери дали даденото лице може да гласува. Ако KEP-а е валиден и лицето има право на глас, то се представя изборното меню. Гласоподавателят гласува и се изпраща заявка, че даденият гласоподавател е гласувал. В друга заявка се изпраща неговият избор. Заявките се изпращат към модула "връзка със сървъра". Те трябва да се обработят атомарно - една след друга. След успешното му изпращане, на екрана на гласоподавателя ще се изпише съобщение за успешно подаден глас.
 - *регионален администратор*
 - *добавяне на председател на секция*
 - *AddChairman(EGN: String, sectionId: int)*
Входни данни: ЕГН на бъдещия председател на секция и код на секция
Изходни данни: съобщение за успешно назначаване на председател в дадената секция
Ограничения при употреба: Лицето, съответстващо на подаденото ЕГН трябва да има правото на глас, кода на

секцията трябва да съществува и тя трябва да бъде свободна

Грешки и изключения: при невалидно или несъществуващо ЕГН, системата извежда съобщение за грешка

Зависимост от други елементи: "връзка със сървър"

Пример за употреба: Регионалният администратор въвежда ЕГН, което бива изпратено към сървъра чрез модула "връзка със сървър". Ако това ЕГН е валидно (тоест то съществува и съответното лице има правото да гласува) системата назначава конкретното лице на председател на подадената секция, като му изпраща имейл, че е получил съответния акаунт. Акаунтът му на председател на секция е със същия имейл и парола като другия, но се влиза в системата с него посредством терминал.

- *гласоподавател*

- *гласуване на място*

- *Vote(choice: int, idCardNumber: string, EGN: string)*

Входни данни: получава се номерът на картата и ЕГН-то на гласоподавателя посредством скенера на личната карта.

Изходни данни: код, показващ за кого е гласът и заявка за маркиране на гласоподавателя като гласувал

Ограничения при употреба: Всеки гласоподавател може да гласува само веднъж в рамките на изборния ден. Също така, тази функция става налична след като бъде сканирана лична карта.

Зависимост от други елементи: "сканиране", "връзка със сървър"

Пример за употреба: След като сканира своята лична карта, ако личната карта е валидна и човекът има право да гласува, се представя изборното меню. Гласоподавателят избира за кого ще гласува и се подава заявката към модула "връзка със сървър". В друга заявка се изпраща, че даденият гласоподавател е гласувал. Двете заявки трябва да се обработят атомарно - една след друга. След успешното изпращане, на екрана на терминала се изписва съобщение за успешно подаден глас.

- *(електронен) потребител*

- *регистрация*

- *Register(email: String, password: String, EGN: String, KEP: String)*

Входни данни: имейл, парола, ЕГН, KEP

Изходни данни: съобщение за успешна регистрация

Ограничения при употреба: всеки човек има право на единствен акаунт в системата

Грешки и изключения: при невалидни входни данни или вече съществуващи такива, регистрацията се отхвърля.

Зависимост от други елементи: “връзка със сървър”

Пример за употреба: Потребител въвежда имейл, парола, ЕГН и КЕП. Те се изпращат към сървър, за да се провери дали такъв акаунт вече съществува в категория “регистриран потребител”. Ако няма такъв, то се изпраща заявка за създаване на акаунт към “връзка със сървър” и след това се връща съобщение за успешна регистрация. Автоматично потребителят влиза в системата като регистриран гласоподавател посредством своя акаунт.

- *главен администратор*

- *въвеждане на данни на кандидат*

- *addCandidate(newCandidate: Candidate)*

Входни данни: данни за кандидат

Изходни данни: съобщение за успешно добавяне на кандидата в изборите

Ограничения при употреба: един кандидат може да бъде записан само веднъж

Грешки и изключения: при подаване на невалиден кандидат (който не е регистриран в системата или не може да гласува) се извежда съобщение за грешка

Зависимост от други елементи: “връзка със сървър”

Пример за употреба: Главен администратор въвежда данните на новия кандидат, те се изпращат към сървър чрез модула “връзка със сървър” и ако той е добавен успешно, се изписва съобщение, че операцията е успешна.

- *публикуване на резултати*

- *PublishResults()*

Изходни данни: няма

Ограничения при употреба: Тази функция ще е достъпна след края на изборния ден

Зависимост от други елементи: “връзка със сървър”

Пример за употреба: След края на изборния ден, главният администратор достъпва тази функция. Изпраща се заявка към сървъра посредством модула “връзка със сървър”, чрез която резултатите стават видими за потребителите на системата.

- *удължаване на изборен ден*

- *ProlongVotingDay(newEndTime: Time, code: int)*

Входни данни: време на приключване на изборния ден (HH:MM) и цялата страна/регион/определена секция

Изходни данни: съобщение за успешно удължено време

Грешки и изключения: при въвеждане на час, преди края на изборния ден се връща съобщение за грешка.

Зависимост от други елементи: “връзка със сървъра”

Пример за употреба: При невъзможност на всички гласоподаватели да гласуват в рамките на изборния ден, главният администратор избира тази опция. Това води до изпращане на заявка до сървър посредством модула “връзка със сървъра”. Ако операцията е успешна се удължава изборния ден до посоченото време.

- *добавяне на регионален администратор*

- *AddRegionalAdmin(EGN: String, regionId: Int)*

Входни данни: ЕГН на бъдещия регионален администратор и код на региона

Изходни данни: съобщение за успешно назначаване на регионален администратор в дадения регион

Ограничения при употреба: Лицето, съответстващо на подаденото ЕГН трябва да има правото на глас, кода на региона трябва да съществува и той трябва да бъде свободен

Грешки и изключения: при невалидно или несъществуващо ЕГН, системата извежда съобщение за грешка

Зависимост от други елементи: “връзка със сървъра”

Пример за употреба: Главният администратор въвежда ЕГН, което бива изпратено към сървъра чрез модула “връзка със сървъра”. Ако това ЕГН е валидно (тоест то съществува и съответното лице има правото да гласува) системата назначава конкретното лице като регионален администратор на подадения регион като му изпраща имейл, че е получил съответния акаунт. Акаунтът му на регионален администратор е със същия имейл и парола като другия, но се влиза в системата като административно лице, не като регистриран гласоподавател.

- *председател на секция*

- *въвеждане на резултати*

- *AddResults(votes: list<int>, numReceivedVotes:int)*

Входни данни: списък с броя гласове, които е получил всеки от кандидатите (въвежда се председателя на секцията)

Изходни данни: списъкът с броя гласове се препраща към “връзка със сървъра”

Ограничения при употреба: заявката може да се осъществи само веднъж и трябва да бъде след края на изборния ден

Грешки и изключения: ако сумата от броя на гласовете, които се изпращат за добавяне е различна от броя сканирани карти от председателя на секцията се връща съобщение за грешка.

Зависимост от други елементи: “връзка със сървър”

Пример за употреба: След края на изборния ден, председателят на дадена секция въвежда списък с броя гласове, които е получил всеки от кандидатите. Ако броят е правилен, то списъкът се препраща към сървър на системата чрез “връзка със сървър”, за да бъде добавен към базата данни.

- *логин*

- *регистриран гласоподавател*

- *LoginAsRegistered(email: String, password: String)*

- Входни данни:** имейл, парола

- Изходни данни:** съобщение за успешно влизане в системата

- Грешки и изключения:** при въведени невалидни данни, влизането в системата се отхвърля

- Зависимост от други елементи:** “връзка със сървър”

- Пример за употреба:** Потребител въвежда имейл и парола. Те се изпращат към сървър чрез модула “връзка със сървър”, където се проверява дали информацията за акаунта е коректна. Това става посредством заявка към базата данни към категорията “регистриран потребител”. Ако данните са коректни, се изпраща съобщение за успешно влизане в системата.

- *административно лице*

- *LoginAsAdmin(email: String, password: String, KEP: String)*

- Входни данни:** имейл, парола, кеп

- Изходни данни:** съобщение за успешно влизане в системата

- Грешки и изключения:** при въведени невалидни данни, влизането в системата се отхвърля

- Зависимост от други елементи:** “връзка със сървър”

- Пример за употреба:** Потребител въвежда имейл, парола и КЕП. Те се изпращат към сървър посредством модула “връзка със сървър”, където се проверява дали информацията за акаунта е коректна. Това става посредством заявка към базата данни към категорията “административни лица”. Ако данните са коректни, се изпраща съобщение за успешно влизане в системата.

- *връзка със сървър*

- *SendQuery(query: Query, type: int)*

- Входни данни:** заявка, която трябва да се препрати към сървър

- Изходни данни:** връща отговор за съответната заявка

- Грешки и изключения:** при неналичност на сървър, да връща съобщение за грешка.

Зависимост от други елементи: всички модули и подмодули от „Потребител“ и „връзка с потребител“ от „сървър“

Пример за употреба: получава криптирана заявка от даден модул и я препраща към сървъра, където тя се получава чрез „връзка с потребител“. След като се обработи, връща отговора към съответния модул, който е изпратил заявката първоначално.

- *сканиране*

- *Scan(idCard: Image)*

Входни данни: сканирана снимка на личната карта на гласоподавател

Изходни данни: номер на картата и ЕГН

Грешки и изключения: при невалидна лична карта или невъзможност на лицето да гласува, се връща съобщение за грешка.

Зависимост от други елементи: „гласуване“, „връзка със сървъра“

Пример за употреба:

- Гласоподавател сканира картата си. Ако сканирането е успешно, той получава възможността да гласува
- Председател на секция сканира лична карта (лични карти). Ако сканирането е успешно, притежателят на личната карта се отбелязва като гласувал. Инкрементира се броячът за гласувалите с хартиена бюлетина.

- *статистика*

- *ShowStatistics(Filters: Filter)*

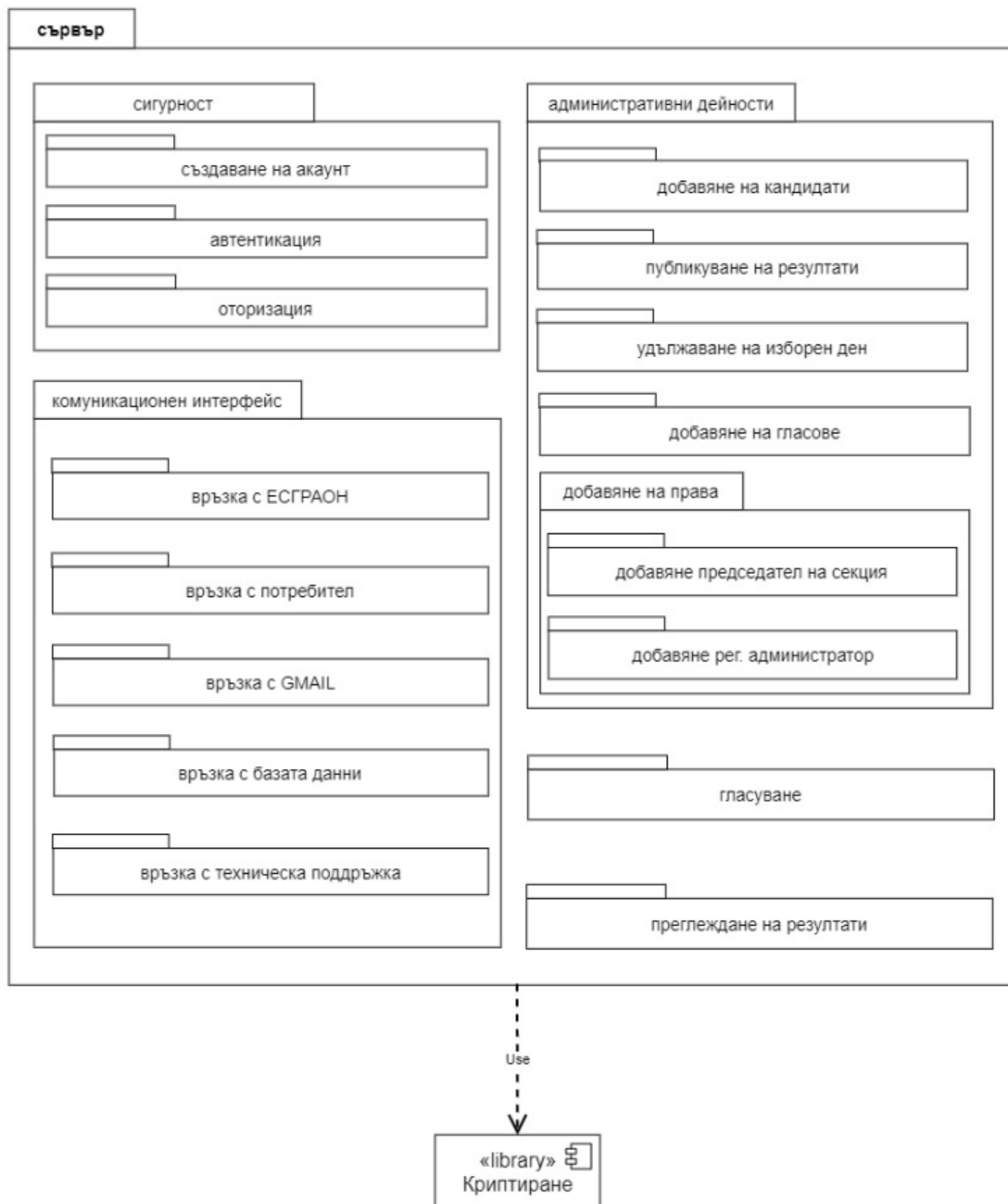
Входни данни: заявка за представяне на резултати

Изходни данни: представяне на желаните резултати

Зависимост от други елементи: „връзка със сървъра“, „профили“

Пример за употреба: Потребител (не задължително логнат в системата) избира опция „статистика“ и избира желаните от него филтри. Чрез сървъра се препраща заявка до базата данни за избори, отговарящи на критериите и желаната от потребителя информация се представя на екрана.

2. Сървър



- **предназначение на модула** - Този модул осъществява връзката между отделните модули с базата данни. Това ще се осъществява чрез получаване на заявки от отделните модули, които ще се обработват и в последствие ще се връща отговор след запитване до базата данни.
- **основни отговорности на модула:**
 - осъществяване на 100% наличност по време на изборния ден
 - осигуряването на сигурен пренос на данни между него и базата данни
 - атомарното обработване и отговаряне на заявките
 - сървърите на системата трябва да могат да издържат на натоварване от поне 100000 заявки по едно и също време
 - сървърите на системата трябва да осигуряват висока производителност - всяка заявка трябва да се обработва за не повече от 1 секунда

Описание на интерфейсите на модула:

- *сигурност*
 - *създаване на акаунт*
 - *CreateAccount(email: String, password: String, EGN: String, KEP: String)*

Входни данни: данни за потребител, получени от модула „връзка с потребител“

Изходни данни: съобщение за успешно/неуспешно добавен акаунт

Зависимост от други елементи: „връзка с потребител“, „връзка с техническа поддръжка“ и „връзка с базата данни“ от „комуникационен интерфейс“ и също така „автентикация“

Пример за употреба: Получават се данни на потребителя, който желае да се регистрира в системата. Те се получават в криптиран вид от „връзка с потребител“ и се декриптират. Изпращат се към „автентикация“, за да бъдат проверени данните за валидност. След това при валидни входни данни, се пращат данните към „връзка с базата данни“, за да бъде създаден акаунт с посочените данни.
 - *автентикация*
 - *authenticateUser(email: String, type: int, password: String, EGN: String = NULL, KEP: String = NULL)*

Входни данни: имейл, тип на потребител, парола и при необходимост КЕП.

Изходни данни: връща отговор дали данните за акаунта са коректни, тоест дали потребителят е този, за който се представя

Зависимост от други елементи: „създаване на акаунт“, „оторизация“ както и всички подмодули от „комуникационен интерфейс“.

Пример за употреба: Получава се информация за даден акаунт, изпраща се заявка към базата данни с цел да се провери дали

данните са коректни. Ако те са, то данните се предават на оторизация.

- *оторизация*
 - *authorizeUser(email: String, type: int)*

Входни данни: получава данните на акаунта, чиито права трябва да се проверят.

Изходни данни: връща списък с правата на потребителя на системата.

Зависимост от други елементи: „автентикация“ както и от „комуникационен интерфейс“ подмодулите „връзка с потребител“, „връзка с техническа поддръжка“ и „връзка с базата данни“

Пример за употреба: Получава се информация за акаунта, чиито права трябва да се определят. За тази цел се изпраща заявка до базата данни. Полученият резултат се препраща се към модула „Потребител“ чрез „връзка с потребител“.
- *комуникационен интерфейс*
 - *връзка с ЕСГРАОН*
 - *connectWithESGRAON(query: Query, type: int)*

Входни данни: заявка, която трябва да се препрати към ЕСГРАОН и код на модула, който я е подал

Изходни данни: отговор на заявката под формата на съобщение, което се изпраща до съответния модул, определен от type.

Грешки и изключения: При неналичност на ЕСГРАОН се връща съобщение за грешка.

Зависимост от други елементи: ЕСГРАОН, „автентикация“, „гласуване“, „добавяне на кандидати“, модула „добавяне на права“ и „връзка с техническа поддръжка“.

Пример за употреба: Получава се заявка, която се препраща към ЕСГРАОН и се връща на съответния модул получения отговор.
 - *връзка с потребител*
 - *connectWithUser(query: Query, type: int)*

Входни данни: заявка, която трябва да се препрати към някой подмодул от сървъра, както и информация от кой подмодул е подадена.

Изходни данни: отговор на заявката под формата на съобщение, което се изпраща до модула, определен от type

Грешки и изключения: При невъзможност да се свърже с потребителя, който е изпратил заявката, процесът се прекратява.

Зависимост от други елементи: всички модули от „Сървър“ без останалите подмодули от „комуникационен интерфейс“ както и „връзка със сървър“ от „Потребител“

Пример за употреба: Получава се заявка от „връзка със сървър“ от „Потребител“, за да бъде обработена. Тя се препраща към

съответния модул, за който е предназначена. Впоследствие връща отговора обратно към „връзка със сървър“ от „Потребител“.

- *връзка с GMAIL*

- *connectWithGMAIL(query: Query, type: int)*

- Входни данни:** заявка, която трябва да се препрати към GMAIL и код на модула, който я е подал

- Изходни данни:** отговор на заявката под формата на съобщение, което се изпраща до съответния модул, определен от type.

- Грешки и изключения:** При неналичност на GMAIL се връща съобщение за грешка.

- Зависимост от други елементи:** „автентикация“, GMAIL, „връзка с техническа поддръжка“

- Пример за употреба:** Получава се заявка, която се препраща към GMAIL и се връща на съответния модул полученият отговор от GMAIL. Тази заявка може да бъде за валидиране на имейл адрес или изпращане на имейл напразно.

- *връзка с базата данни*

- *connectWithDatabase(query: Query, type: int)*

- Входни данни:** заявка, която трябва да се препрати към базата данни, както и информация от кой модул е подадена

- Изходни данни:** отговор на заявката под формата на съобщение, което се изпраща до модула, определен от type

- Грешки и изключения:** При невъзможност да се свърже с базата данни, се изчаква тя да бъде отново налична.

- Зависимост от други елементи:** всички модули от „Сървър“ без останалите подмодули от „комуникационен интерфейс“

- Пример за употреба:** Получава се заявка, която се препраща към „Мениджъра на базата данни“, който ще я обработи и ще изпрати отговор. Този отговор се получава от „връзка с базата данни“ и се препраща към модула, който е изпратил.

- *връзка с техническа поддръжка*

- *connectWithTechnicalSupport(query: Query, type: int)*

- Входни данни:** заявка, която ще се препрати към „техническа поддръжка“ както и информация от кой модул е изпратена.

- Изходни данни:** отговор на заявката под формата на съобщение, което се изпраща до модула, определен от type

- Грешки и изключения:** при неналичност на „техническа поддръжка“ операцията се прекратява.

- Зависимост от други елементи:** всички модули и подмодули от „Сървър“ без „връзка с потребител“ и „връзка с базата данни“

- Пример за употреба:** Получава се заявка от „връзка със сървър“ от модула „техническа поддръжка“, която се препраща към съответния модул от „Сървър“. След като се обработи заявката се изпраща обратно отговорът, получен от съответния модул.

- *преглеждане на резултати*

- *ShowStatistics(Filters: Filter, userType: int)*

Входни данни: получава се заявката за резултати от избори както и типа потребител, който я е изпратил.

Изходни данни: Данните, които трябва да се представят се изпращат към "връзка с потребител"

Зависимост от други елементи: „връзка с потребител“, „връзка с базата данни“, „връзка с техническа поддръжка“.

Пример за употреба: Получава се заявка от потребител на системата за преглеждане на резултати от дадени избори. Ако потребителят има права да ги преглежда (резултатите са публикувани или потребителят е администратор, който има право да ги преглежда преди официалното им обявяване), то се изпраща заявка до базата данни за извличане на търсените резултати. Те се представят на потребителя.

- *административни дейности*

- *добавяне на кандидати*

- *addCandidate(newCandidate: Candidate)*

Входни данни: данни за кандидат

Изходни данни: изпраща съобщение за успешно добавяне на кандидат към „връзка с потребител“

Ограничения при употреба: един кандидат може да бъде записан само веднъж

Грешки и изключения: при подаване на невалиден кандидат (който не е регистриран в системата или не може да гласува) се изпраща съобщение за грешка

Зависимост от други елементи: ЕСГРАОН, всички подмодули от „комуникационен интерфейс“ без „връзка с GMAIL“.

Пример за употреба: От "връзка с потребител" се получава информация за кандидат, който трябва да бъде добавен. Той се добавя чрез заявка до базата данни и се връща съобщение дали тази операция е успешна. Това съобщение се препраща обратно към "връзка с потребител".

- *публикуване на резултати*

- *PublishResults()*

Изходни данни: Няма

Ограничения при употреба: Тази функция ще е достъпна след края на изборния ден и само за главния администратор.

Зависимост от други елементи: "връзка с базата данни", "връзка с потребител", "връзка с техническа поддръжка".

Пример за употреба: От "връзка с потребител" се получава заявка за показване на резултатите. Изпраща се заявка към "връзка с базата данни" да ги направи видими, след това се препраща съобщение, че операцията е завършила успешно.

- *удължаване на изборен ден*
 - *ProlongVotingDay(newEndTime: Time, code: int)*
Входни данни: време на приключване на изборния ден (HH:MM) от “връзка с потребител” за цялата страна/регион/определена секция
Изходни данни: съобщение за успешно удължено време, което се препраща към “връзка с потребител”
Грешки и изключения: при въвеждане на час, преди края на изборния ден се изпраща съобщение за грешка към “връзка с потребител”
Зависимост от други елементи: “връзка с базата данни”, „връзка с потребител“, „връзка с техническа поддръжка“.
Пример за употреба: Получава се ново време от модула “връзка с потребител” за приключване на изборния ден за цялата страна/регион/определена секция, което да бъде приложено за края на изборния ден.
- *добавяне на гласове*
 - *AddVotes(votes: list<int>)*
Входни данни: списък с броя гласове, които е получил всеки от кандидатите, получени от декриптирана заявка от “връзка с потребител”
Изходни данни: изпраща се съобщение към “връзка с потребител” за успешно добавяне на гласовете в базата данни.
Ограничения при употреба: Този вид заявки може да се осъществяват след края на изборния ден.
Грешки и изключения: ако сумата от броя на гласовете, които се изпращат за добавяне е различна от броя сканирани карти от председателя на секцията се връща съобщение за грешка.
Зависимост от други елементи: “връзка с базата данни”, „връзка с потребител“, „връзка с техническа поддръжка“.
Пример за употреба: Получава се списък с броя гласове, които е получил всеки от кандидатите, получени от “връзка с потребител”. Те се изпращат чрез подмодула “връзка с базата данни” към базата данни. Тя ще ги добави към текущите резултати за изборите.
- *добавяне на права*
 - *добавяне на председател на секция*
 - *addChairman(EGN: String, sectionId: int)*
Входни данни: ЕГН на бъдещия регионален администратор и код на региона от “връзка с потребител”
Изходни данни: препраща съобщение за успешно назначаване на председател в дадената секция към “връзка с потребител”
Ограничения при употреба: Лицето, съответстващо на подаденото ЕГН трябва да има правото на глас, кода на

секцията трябва да съществува и тя трябва да бъде свободна

Грешки и изключения: при невалидни входни данни, системата препраща съобщение за грешка към “връзка с потребител”

Зависимост от други елементи: “връзка с базата данни”, „връзка с потребител“, „връзка с техническа поддръжка“ и „връзка с ЕСГРАОН“

Пример за употреба: От подмодула “връзка с потребител” се получава заявка за добавяне на председател на секция. Проверява се дали подадената секция е свободна чрез заявка към базата данни. След това се установява дали такъв потребител с такова ЕГН съществува в базата данни на системата и ако да, се изпраща заявка към ЕСГРАОН да се провери дали лицето има правото на глас.

- *добавяне на регионален администратор*

- *addRegionalAdmin(EGN: String, regionId: int)*

Входни данни: ЕГН на бъдещия регионален администратор и код на региона от “връзка с потребител”

Изходни данни: препраща се съобщение за успешно назначаване на регионален администратор в дадения регион към “връзка с потребител”

Ограничения при употреба: Лицето, съответстващо на подаденото ЕГН трябва да има правото на глас, кода на региона трябва да съществува и той трябва да бъде свободен

Грешки и изключения: при невалидно или несъществуващо ЕГН, се препраща съобщение за грешка

Зависимост от други елементи: “връзка с базата данни”, „връзка с потребител“, „връзка с техническа поддръжка“ и „връзка с ЕСГРАОН“.

Пример за употреба: От подмодула “връзка с потребител” се получава заявка за добавяне на регионален администратор. Проверява се дали подадения регион е свободен чрез заявка към базата данни. След това се установява дали такъв потребител с такова ЕГН съществува в базата данни на системата и ако да, се изпраща заявка към ЕСГРАОН да се провери дали лицето има правото на глас.

- *гласуване*

- *ValidateVote(KEP: String = NULL, idCardNumber: int = NULL)*

Входни данни: КЕП на гласоподавател или номер на лична карта от “връзка с потребител”.

Исходни данни: съобщение за успешна/неуспешна валидация на съответното лице, което се изпраща към „връзка с потребител“.

Зависимост от други елементи: „връзка с базата данни“, „връзка с потребител“, „връзка с техническа поддръжка“ и „връзка с ЕСГРАОН“.

Пример за употреба: Получават се различни входни данни, в зависимост от това кой е извикал този модул. След това се изпраща заявка към „връзка с ЕСГРАОН“, да се потвърди даденото лице (представено или чрез КЕП, или чрез номер на лична карта) дали може да гласува. В случай, че той може, се изпраща още една заявка към „връзка с базата данни“ да се провери дали даденото лице не е гласувало вече.

- *AddVote(choice: int)*

Входни данни: код на избрания кандидат

Исходни данни: изпраща се съобщение към „връзка с потребител“ за успешно отчетен глас

Зависимост от други елементи: „връзка с базата данни“, „връзка с потребител“ и „връзка с техническа поддръжка“.

Пример за употреба: Получава се код на избран кандидат от „връзка с потребител“, след което този код бива изпратен към „връзка с базата данни“, за да бъде записан в базата данни.

- *MarkAsVoted(KEP: String = NULL, idCardNumber: int = NULL)*

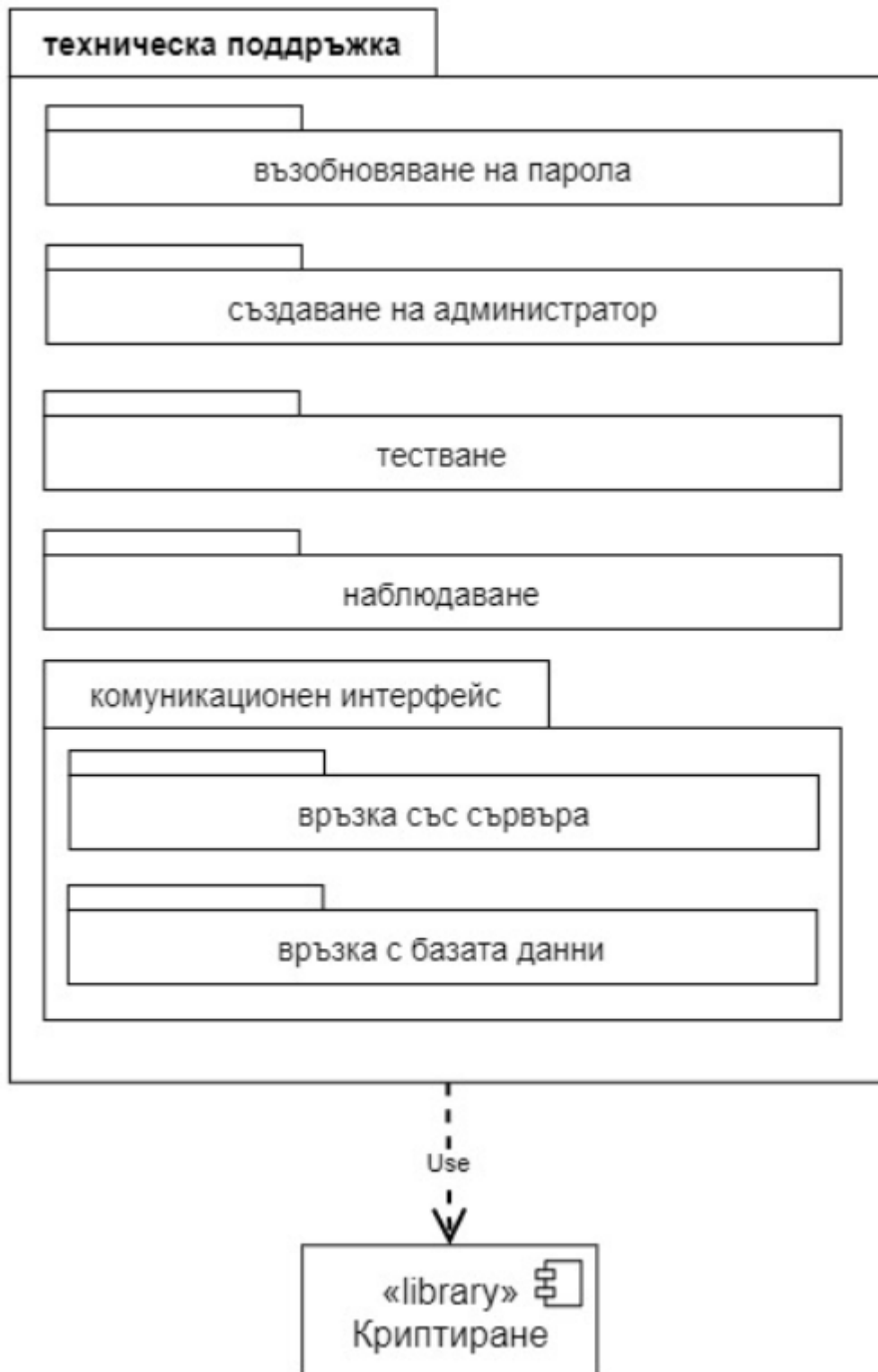
Входни данни: КЕП на гласоподавател

Исходни данни: изпраща се съобщение към „връзка с потребител“ за успешно отбелязан гласоподавател

Зависимост от други елементи: „връзка с базата данни“, „връзка с потребител“ и „връзка с техническа поддръжка“.

Пример за употреба: Получава се КЕП/номер на лична карта на гласоподавател от „връзка с потребител“. След което данните биват изпратени към „връзка с базата данни“, за да бъде отбелязан съответния гласоподавател като гласувал.

3. Техническа поддръжка



- **предназначение на модула** - Този модул позволява на екипа по техническа поддръжка да осъществява дейността си и по този начин да се подsigури коректността на предоставяните от системата услуги.
- **основни отговорности на модула:**
 - тестване на системата
 - мониторинг на системата
 - допълнителни дейности (създаване на администратор и възобновяване на парола), за да се осигури нормален начин на работа на системата

Описание на интерфейсите на модула:

- **възобновяване на парола**
 - *createNewPassword(KEP: String)*
Входни данни: КЕП, чрез който регистрираният потребител доказва своята самоличност
Изходни данни: съобщение за успешно възобновяване на паролата
Грешки и изключения: Ако КЕП-а е невалиден, се връща съобщение за грешка.
Ограничения при употреба: Функцията се разрешава да се приложи за дадения регистриран гласоподавател, само след като екипът по техническа поддръжка разгледа внимателно случая, за да е сигурен, че не е измама
Зависимост от други елементи: „връзка със сървър“ и „връзка с базата данни“
Пример за употреба: Получава се заявка от регистриран гласоподавател за възобновяване на паролата. След като се установи, самоличността на потребителя, чрез съответните заявки до базата данни, се осъществява операцията.
- **създаване на администратор**
 - *createAdmin(email: String, EGN: String)*
Входни данни: Въвежда се имейлът и ЕГН на регистрираният гласоподавател, който ще бъде главен администратор.
Изходни данни: съобщение, че главният администратор е добавен успешно.
Ограничения при употреба: За дадени избори, главният администратор трябва да е 1.
Зависимост от други елементи: „връзка с базата данни“, „връзка със сървър“
Пример за употреба: Преди изборите, се определя кой ще бъде главен администратор и му се създава акаунт посредством неговият имейл и КЕП. След като се създаде акаунта, се връща съобщение, че операцията е приключила успешно.

- *тестване*

- *runTest(newTest: test)*

Входни данни: получават се тестовете, които трябва да бъдат изпълнени.

Изходни данни: връщат се резултатите от осъществените тестове.

Зависимост от други елементи: „връзка с базата данни“ и „връзка със сървър“

Пример за употреба: Екипът от техническа поддръжка решава, че трябва да тества дадена функционалност. Модулът извършва тестовете и връща резултатите от тях.

- *наблюдаване*

- *monitor(modulToMonitor: int)*

Входни данни: получава се номер, по който да се покаже кой модул ще се наблюдава.

Изходни данни: връщат се резултатите от проведените наблюдения на модулите на системата

Зависимост от други елементи: : „връзка с базата данни“ и „връзка със сървър“.

Пример за употреба: Екипът от техническа поддръжка желае да наблюдава даден модул. Изпраща се съответната заявка към „комуникационен интерфейс“ и се връщат резултатите от наблюденията, когато те завършат.

- *комуникационен интерфейс*

- *връзка със сървър*

- *connectWithServer(query: Query, type: int)*

Входни данни: заявка, която трябва да се препрати към модул от сървър, както и информация от кой модул е подадена.

Изходни данни: отговор на заявката под формата на съобщение, което се изпраща до модула, определен от type.

Грешки и изключения: При невъзможност да се свърже с сървър, се връща съобщение за грешка и процесът се прекратява.

Зависимост от други елементи: „връзка с техническа поддръжка“ от сървър и от „техническа поддръжка“ – „възобновяване на парола“, „тестване“ и „наблюдение“.

Пример за употреба: Получава се заявка, която се препраща към съответния модул, който да я обработи. След това получава отговор от него, който отговор се препраща към подмодула, който е изпратил заявката.

- *връзка с базата данни*

- *connectWithDatabase(query: Query, type: int)*

Входни данни: заявка, която трябва да се препрати към подмодул от мениджъра на базата данни, както и информация от кой подмодул е подадена

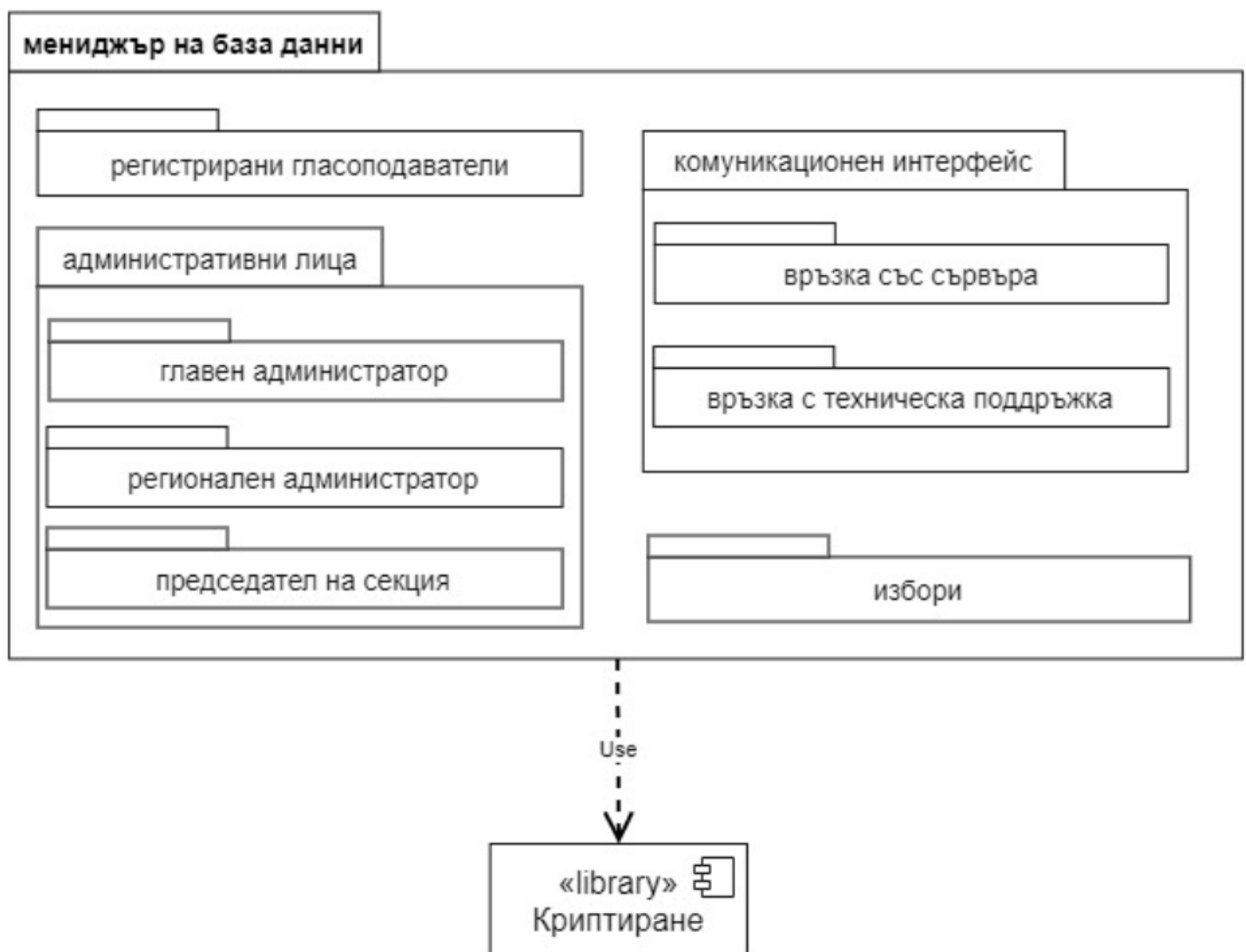
Исходни данни: отговор на заявката под формата на съобщение, което се изпраща до подмодула, определен от type

Грешки и изключения: При невъзможност да се свърже с базата данни, се връща съобщение за грешка и процесът се прекратява.

Зависимост от други елементи: „връзка с техническа поддръжка“ от „Мениджъра на базата данни“ и всички модули от „Техническа поддръжка“ без „връзка сървър“

Пример за употреба: Получава се заявка. Тя се препраща към съответния модул, който да я обработи. След това се получава отговор от този модул. Този отговор се препраща към модула, който е изпратил заявката.

4. Мениджър на база данни

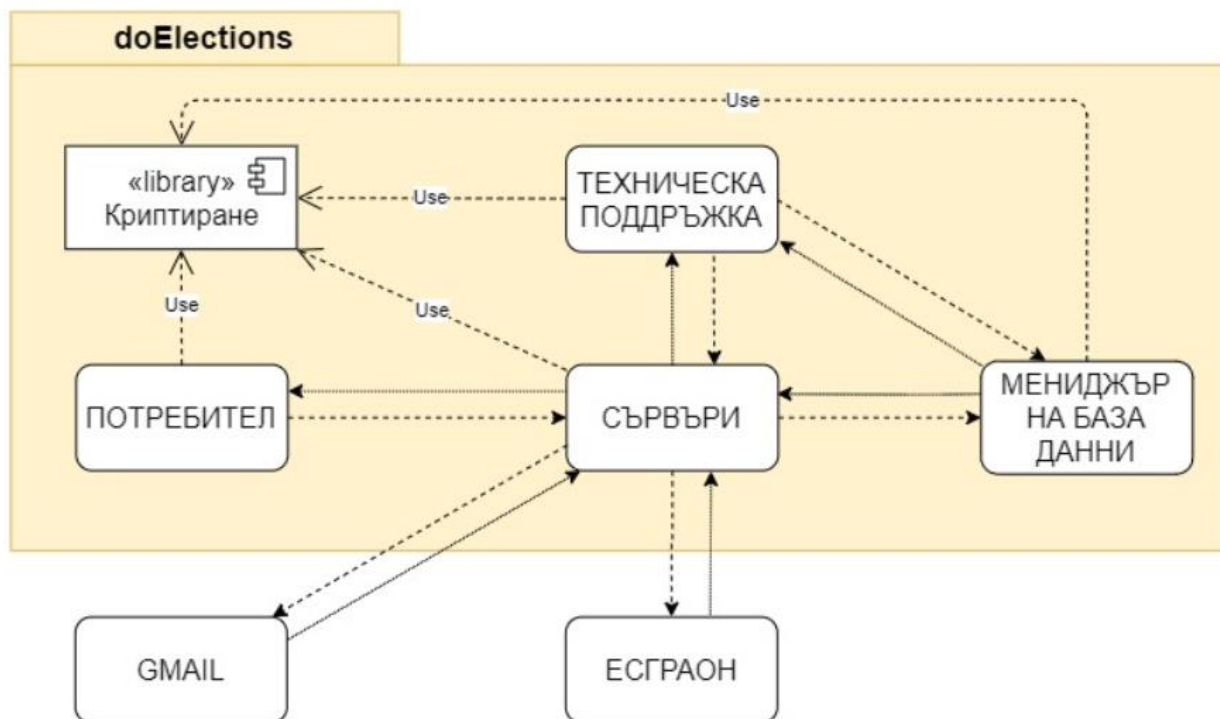


- **предназначение на модула** – Позволява съхраняването и групирането на данните от системата doElections. Това дава възможност както да се осъществят изборите така и да се поддържа статистика за тях.
- **основни отговорности на модула и неговите подмодули:**
 - В „регистрирани гласоподаватели“ и „административни лица“ се съхранява информация за акаунтите в системата на съответния вид потребители.
 - В „избори“ се съхраняват всички нужни данни за изборите. Запазва се информация за това кои са гласували и колко гласа е получил всеки от кандидатите. Също така се пази информация кой е бил главен администратор, на всеки от регионите кой е регионален администратор и за всяка секция – председателят на секцията.
 - „комуникационен интерфейс“ осигурява връзката на модула със системата doElections.

3. Описание на допълнителни структури

3.1 Структура на употреба на модули

3.1.1 Първично представяне



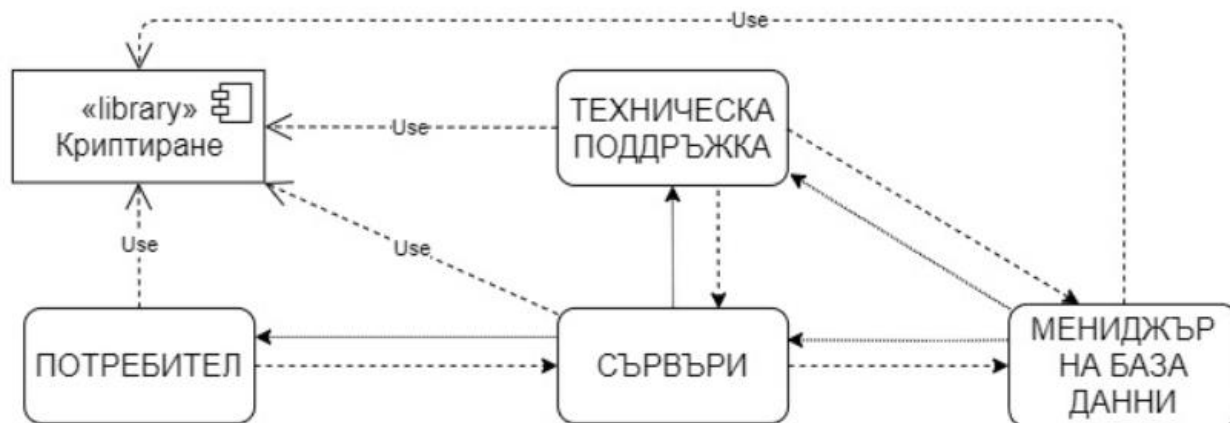
На представената схема виждаме начина, по който се осъществява комуникацията между отделните модули, изграждащи системата doElections.

В тази схема са използвани 3 вида стрелки, с цел по-точно представяне на връзките между модулите. С пунктир се означава посоката, в която комуникацията се инициира, докато вторият вид черта – с пунктир с точки, представлява посоката, в която се връща отговорът към съответната част от системата. Третият вид стрелка – „use“ се използва с цел да се покаже, че всеки от модулите на системата doElections ще използва библиотеката „криптиране“. Тази библиотека се използва както за криптирането така и за декриптирането на данните. Ето защо би било уместно да се представи и със още 1 стрелка от библиотеката „криптиране“ към останалите модули. Това е пропуснато с цел по-добра четимост и прегледност на горе-представената схема.

3.1.2. Описание на елементите и връзките

В тази част ще разгледаме елементите на структурата „употреба на модули“ – модулите, които вече бяха представени в декомпозицията на модули, заедно с връзките помежду им. Тези връзки показват другите модули, от които разглеждания модул (подмодул) пряко се нуждае, за да осъществи нормалната си работа.

3.1.2.1. Връзки между основните модули в системата



3.1.2.1.1. Връзка между Потребител и Сървър

От „Потребител“ информацията се препраща към „Сървър“ посредством модула „връзка със сървъра“. Тя се получава от него чрез „връзка с потребител“. Изпращането на отговор от „Сървър“ към „Потребител“ става по обратния начин – от „връзка с потребител“ от „Сървър“ към „връзка със сървър“ от „Потребител“.

3.1.2.1.2. Връзка между **Сървър** и **Мениджър на база данни**

От „Сървър“ се изпраща запитване към базата данни посредством „връзка с базата данни“. Информацията се получава от „Мениджър на базата данни“ посредством „връзка със сървъра“. След като той обработи запитването се връща отговор по обратния път чрез „връзка със сървъра“ към „връзка с базата данни“ от „сървър“. След като се получи отговора той се декодира и обработва от сървъра.

3.1.2.1.3. Връзка между **Сървър** и **Техническа поддръжка**

От „Техническа поддръжка“ изпращат запитване към „Сървър“ чрез „връзка със сървъра“. То се получава в „Сървър“ от „връзка с техническа поддръжка“ и след като се обработи (проверява се дали сървърът е наличен, осъществяват се тестове или се събират данни за поведението на системата) се изпраща обратно към „Техническа поддръжка“ и се получава от „връзка със сървъра“.

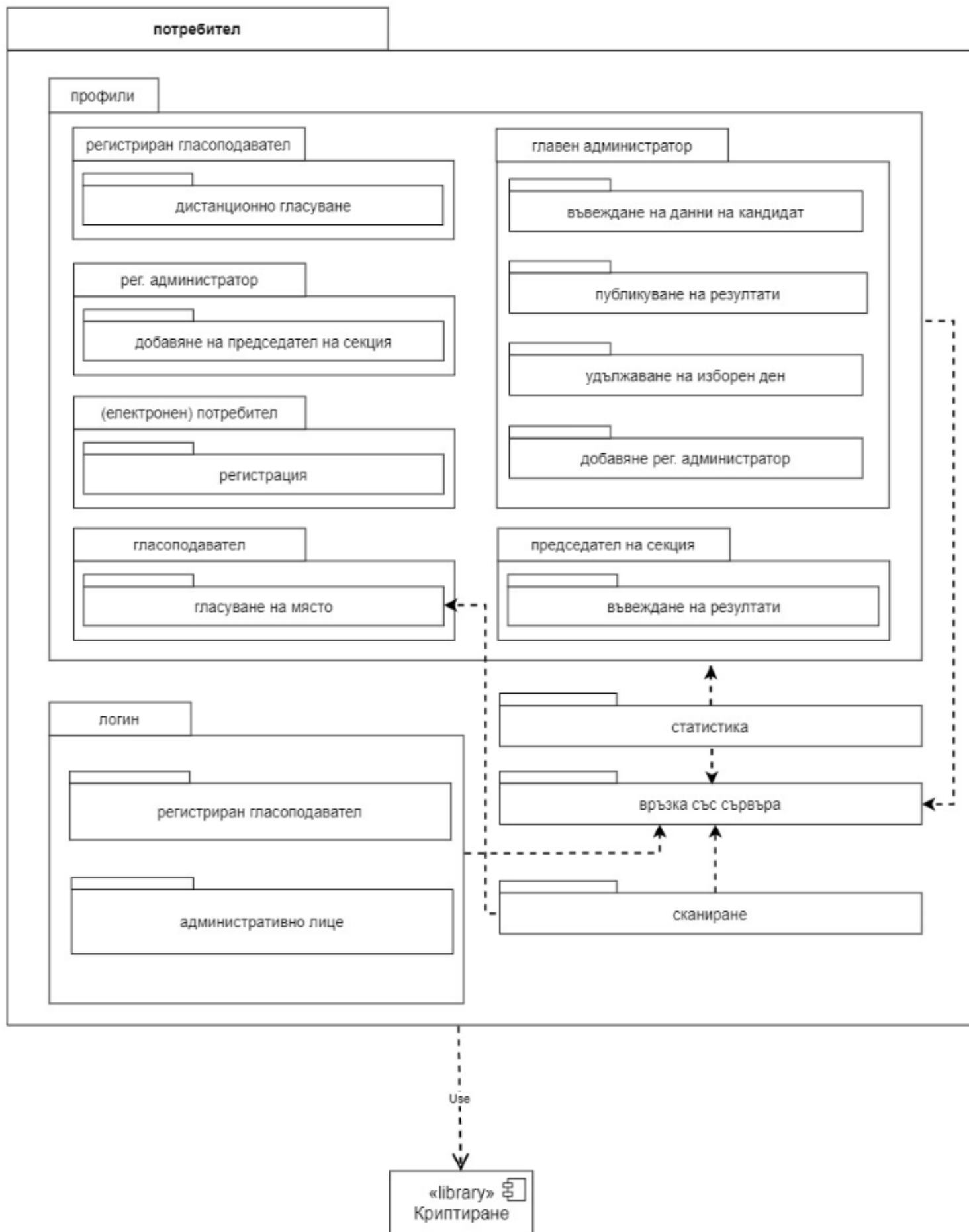
3.1.2.1.4. Връзка между **Техническа поддръжка** и **Мениджър на база данни**

От „Техническа поддръжка“ може да се изпращат заявки и към „Мениджъра на база данни“. Това става чрез „връзка с базата данни“ от „Техническа поддръжка“, а информацията се получава в базата данни посредством „връзка с техническа поддръжка“. Като запитването се приеме, то се обработва и след това се изпраща обратно към „Техническа поддръжка“ чрез „връзка с техническа поддръжка“ и се получава от „връзка с базата данни“.

3.1.2.2. Връзки в основните модули на системата

С цел да се повиши четимостта на схемите, в тази секция ще бъдат пропуснати стрелките, съответстващи на посоката на отговор и ще се концентрираме само върху тези, съответстващи на посоката, в която комуникацията се инициира. На следващите схеми са представени основните връзки в рамките на разглеждания модул и също така е описано тяхното предназначение. Важно е да се отбележи, че на схемите на някои места връзките са „обобщени“. Ако даден модул А включва в себе си други подмодули и е свързат с друг модул В, то това означава, че всеки от подмодулите на А се нуждае от В при нормалната си работа.

3.1.2.2.1. Потребител



3.1.2.2.1.1. Връзка между **сканиране** и **гласуване на място**

Когато гласоподавател желае на да даде своя вот в секция, той трябва да сканира първо личната си карта и след това да избере опцията да гласува.

3.1.2.2.1.2. Връзка между **сканиране** и **връзка със сървър**

По време на изборен ден, председателят на секцията трябва да сканира личните карти на всички хора, желаещи да гласуват с хартиена бюлетина на място. Когато председателят на секция сканира лична карта, информацията се изпраща след криптиране към „връзка със сървър“ и в последствие към „Мениджъра на база данни“, за да се отбележи дадения гласоподавател като гласувал.

3.1.2.2.1.3. Връзка между **статистика** и **профили**

„Статистика“ предоставя различни филтри и по този начин възможност за предоставяне на различна информация в зависимост от вида на потребителя, който прази заявката.

3.1.2.2.1.4. Връзка между **статистика** и **връзка със сървър**

След като потребителят на системата doElections посочи в какви данни е заинтересован, „Статистика“ препраща неговият избор към „връзка със сървър“. В последствие, когато се върнат резултатите от базата данни, те се получават от „статистика“ и представят на потребителя в искания от него вид.

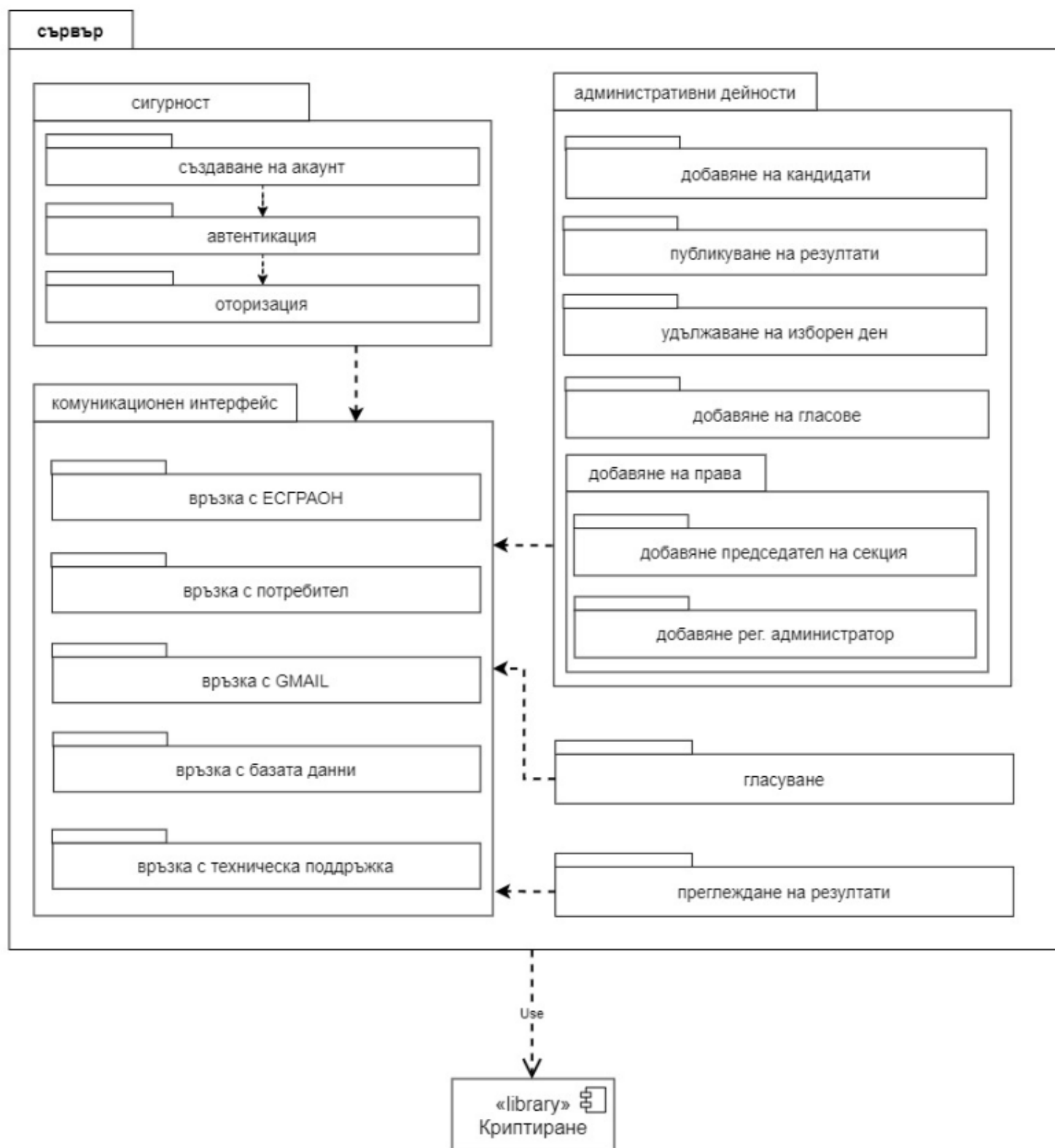
3.1.2.2.1.5. Връзка между **логин** и **връзка със сървър**

Потребител на системата избира като какъв потребител иска да влезе в системата и въвежда изискваните данни, в зависимост от този избор. След като ги въведе, те се изпращат към „връзка със сървър“, за да бъдат препратени за валидиране впоследствие. В зависимост от резултата от валидирането на данните, системата реагира по коректния начин.

3.1.2.2.1.6. Връзка между **профили** и **връзка със сървър**

Всеки един от подмодулите на профили изисква интеракция на системата с дадения вид потребител. Това може да означава както просто избиране на дадена опция в някои случаи, така и много често е необходимо да се въведат данни, да се избират опции и т.н. По този начин се разбира какво точно желае потребителят на системата и когато се получи нужната информация, тя се препраща към „връзка със сървър“ и по този начин се изпраща към „Сървър“, за да бъде обработена. По-късно се получава отговора от „връзка със сървър“ и се връща към съответния подмодул.

3.1.2.2 Сървър



3.1.2.2.2.1. Връзка между **сигурност** и **комуникационен интерфейс**

„Създаване на акаунт“ получава въведените от потребителя данни в случай, че лицето желае да участва в системата като регистриран гласоподавател. Преди акаунтът да бъде създаден (чрез заявка до базата данни чрез „комуникационен интерфейс“) първо се валидират полчените данни чрез „Автентикация“.

„Автентикация“ може да изпраща заявки чрез съответните подмодули до базата данни и ЕСГРАОН, за да се определи дали потребителят наистина е този, за който се представя. „Автентикация“ може да получава данните, които трябва да се проверят както от „създаване на акаунт“ при регистрацията на потребител, така и от „връзка с потребител“ при влизане в системата на потребител. „Оторизация“ след обработване на информацията, връща отговор за това какви права реално притежава дадения потребител на системата.

3.1.2.2.2.2. Връзка между **създаване на акаунт** и **автентикация**

След като се получат данните за акаунта на регистриран гласоподавател, който трябва да се създаде, те се обработват и препращат към „автентикация“ за валидиране. Тя връща отговор дали те са валидни и ако са, се препращат след криптиране посредством „връзка с базата данни“ към „Мениджър на базата данни“, за да бъде добавен новия акаунт в системата. След това се връща отговор дали добавянето е успешно и информацията се препраща към „автентикация“ отново, за да се продължи като при нормално влизане (log in) в системата.

3.1.2.2.2.3. Връзка между **автентикация** и **оторизация**

В „автентикация“ след като се проверят за коректност получените данни и сме сигурни, че потребителят не се опитва да заблуди системата, данните се препращат към „оторизация“. По този начин се определят какви права реално има даденият потребител и след като това стане (и чрез запитвания към други модули) се връща отговор обратно към „връзка с потребител“.

3.1.2.2.2.4. Връзка между **административни дейности** и **комуникационен интерфейс**

В „административни дейности“ са имплементирани редица функционалности, на които се основа коректното администриране на изборите. Всяка една от тях получава от „връзка с потребител“ данни, изпратени от съответното административно лице. Тази информация се обработва и след това се връща отговор към „връзка с потребител“, за да бъде препратена към съответния модул, чрез който е иницирано използването на дадената услуга. Разбира се, при обработването на данните, се изпращат и заявки към базата данни, ЕСГРАОН и при нужда „Техническа поддръжка“.

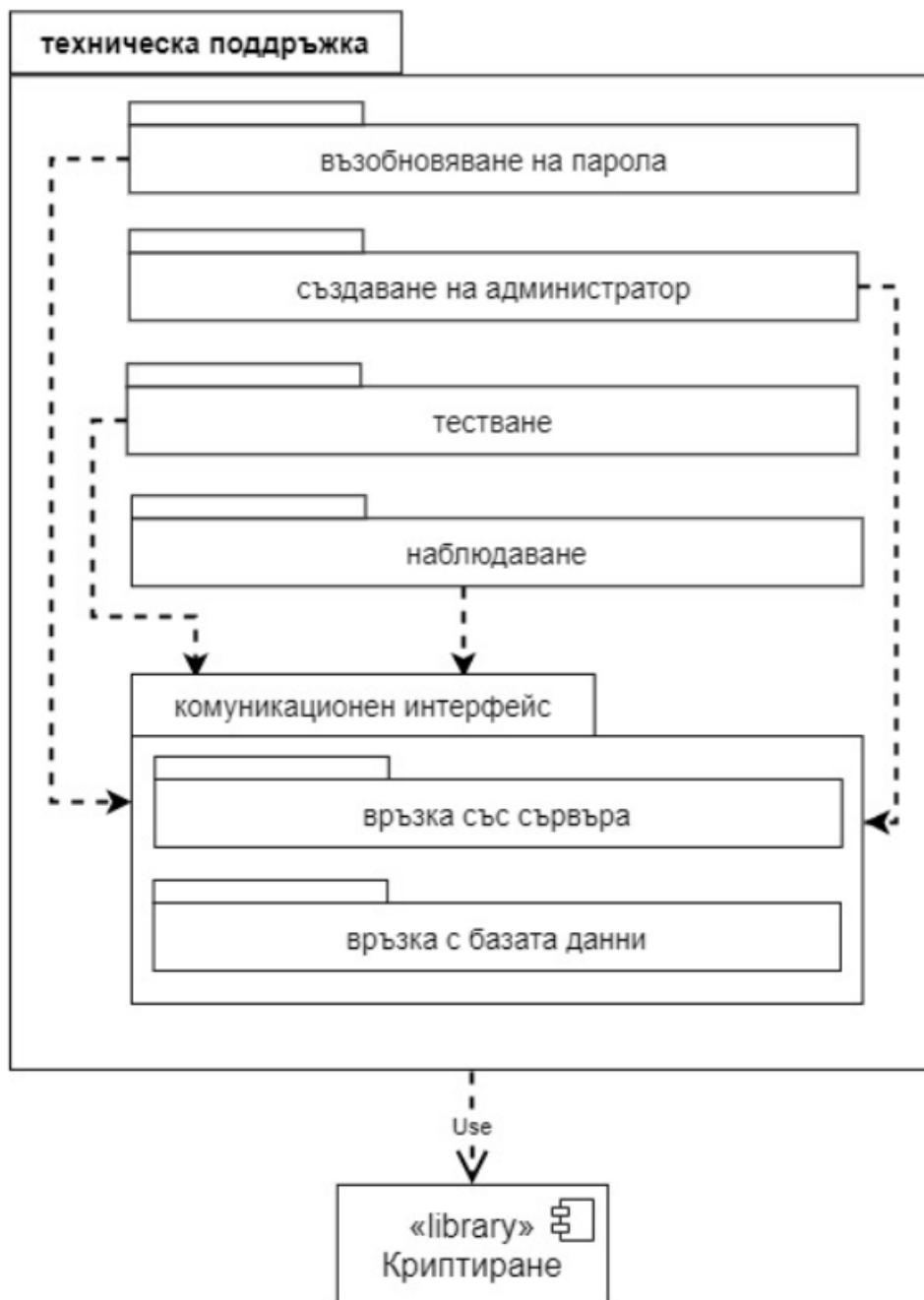
3.1.2.2.2.5. Връзка между **преглеждане на резултати** и **комуникационен интерфейс**

Чрез „връзка с потребител“ се получава заявката на даден потребител за резултати от определени избори. Съгласно получените филтри се изпраща запитване до базата данни за търсените резултати. То се връща, декриптира и след подходящата обработка, чрез „комуникационен интерфейс“ се връща на съответния потребител.

3.1.2.2.2.6. Връзка между **гласуване** и **комуникационен интерфейс**

Чрез „връзка с потребител“ се получават данните на потребителите, които желаят да гласуват. Осъществяват се дейности по валидиране на възможността им да гласуват, чрез запитване до базата данни и ЕСГРАОН. След това се получават данни на хората, които са гласували и също така отделно се получава информация за кого са постъпили гласове. Изпращат се заявки чрез „връзка с базата данни“ за отбелязване на съответните гласоподаватели като вече гласували и също така за отчитане на техните гласове.

3.1.2.2.3. Техническа поддръжка



3.1.2.2.3.1. Връзка между възобновяване на парола и комуникационен интерфейс

„Възобновяване на парола“ получава от „връзка със сървъра“ запитването от страна на потребител за възстановяване на изгубена парола. След като се е установило, че потребителят е истинският притежател на дадения акаунт му се предоставя възможност да въведе нова парола. След това „възобновяване на

парола“ препраща чрез „връзка с базата данни“ новата парола, за да се осъществи промяната.

3.1.2.2.3.2. Връзка между **създаване на администратор и комуникационен интерфейс**

След като се реши кой ще бъде главният администратор за дадените избори, екипът по поддръжка въвежда в „създаване на администратор“ данните му. След това те се криптират и изпращат към базата данни. Също така, при нужда чрез „връзка със сървър“ се правят нужни проверки или главният администратор се известява за назначаването.

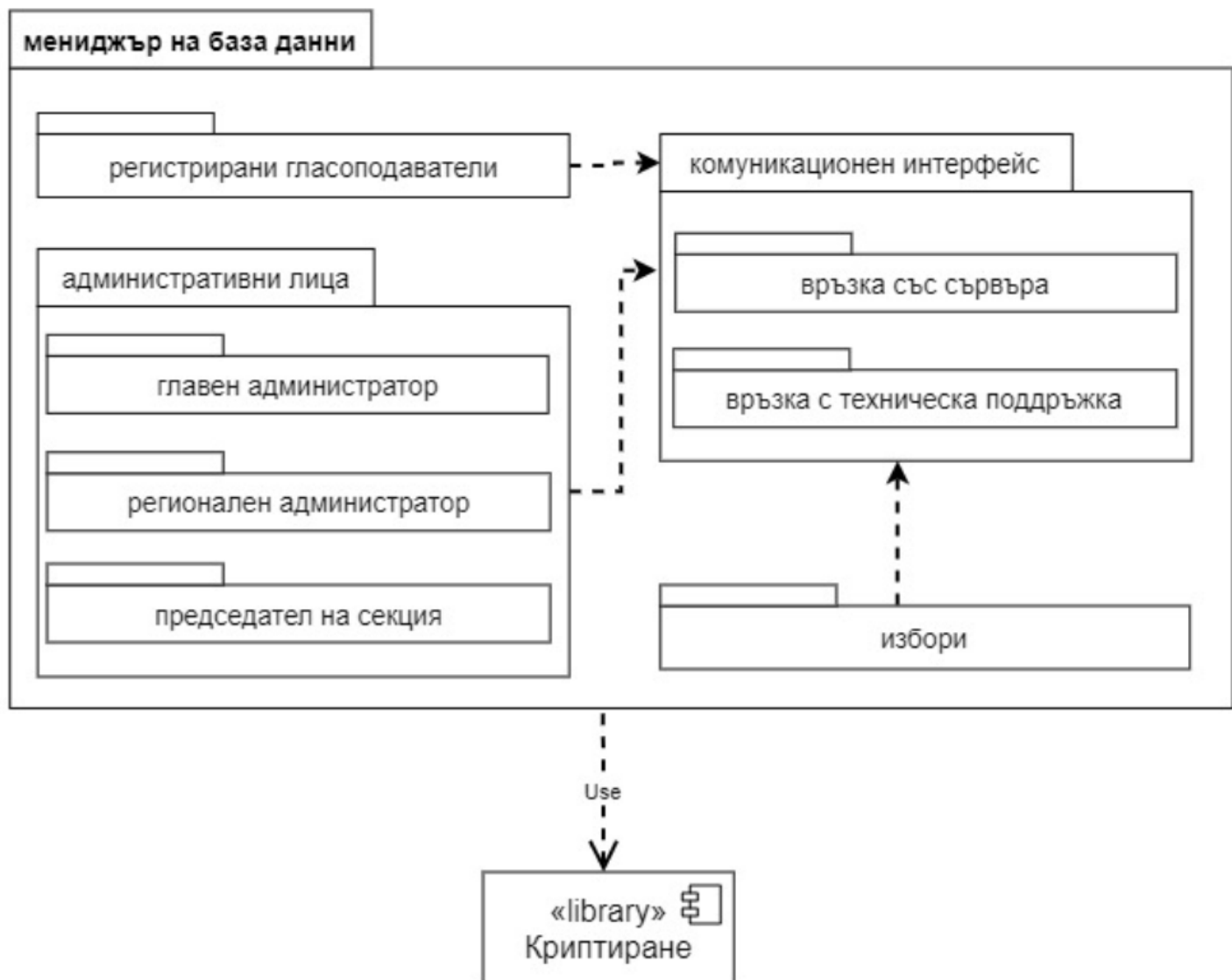
3.1.2.2.3.3. Връзка между **тестване и комуникационен интерфейс**

Чрез „тестване“ екипът по поддръжката може да посочи какво, как и кога ще се тества. След като се посочи тази информация, тя се криптира и препраща към съответния модул, за да се осъществят тестовете. Впоследствие чрез получените чрез „комуникационен интерфейс“ резултати се декриптират и се представят на екипа по поддръжка.

3.1.2.2.3.4. Връзка между **наблюдаване и комуникационен интерфейс**

Чрез „наблюдаване“, екипът по поддръжка може да въведе какво и в какъв времеви интервал да се наблюдава и да му се изследва поведението. Тази информация се изпраща чрез „комуникационен интерфейс“ към съответния модул, за който е предназначена. Впоследствие отново чрез „комуникационен интерфейс“ се връща резултата от проведените наблюдения на системата.

3.1.2.2.4. Мениджър на база данни



3.1.2.2.4.1. Връзка между **регистрирани гласоподаватели** и **комуникационен интерфейс**

След като се получат заявките чрез „комуникационен интерфейс“ и се декриптира, се препраща към „регистрирани гласоподаватели“ тези заявки, които са свързани с акаунтите на регистрираните в системата потребители. Отговорът на тези заявки се връща обратно към „комуникационен интерфейс“.

3.1.2.2.4.2. Връзка между **административни лица** и **комуникационен интерфейс**

„Комуникационен интерфейс“ получава и изпраща към „административни лица“ заявки, които са свързани с акаунти на хора, изпълняващи административни

дейности. Отговорът на тези заявки се връща обратно към „комуникационен интерфейс“.

3.1.2.2.4.3. Връзка между **избори** и **комуникационен интерфейс**

„Комуникационен интерфейс“ изпраща към „избори“ само тези заявки, които са свързани със статистика от минали или текущи избори, информация за това кои са гласували на текущите избори и т.н. Отговорите се връщат към „Комуникационен интерфейс“, за да бъдат криптирани и след това върнати към съответния модул, който се нуждае от исканата информация.

3.1.3 Описание на обкръжението

Както е показано на първичното представяне на употребата на модули, системата doElections е зависимо основно от 2 външни системи – ЕСГРАОН и GMAIL.

3.1.3.1. Връзка на системата с **ЕСГРАОН**

Осъществява се посредством „връзка с ЕСГРАОН“ от „Сървър“. Тази част от „комуникационен интерфейс“ отговаря за изпращане на заявки към ЕСГРАОН, както и тяхното последващо получаване. Системата doElections прави запитвания към ЕСГРАОН в следните ситуации:

3.1.3.1.1. Проверява се самоличността на лицето, чрез ЕГН и номер на лична карта.

3.1.3.1.2. Проверява се дали лице, което желае да гласува има право на глас.

3.1.3.1.3. Проверява се дали лице, което ще бъде председател на секция има право на глас.

3.1.3.1.4. Проверява се дали лице, което ще бъде регионален администратор има право на глас.

3.1.3.2. Връзка на системата с GMAIL

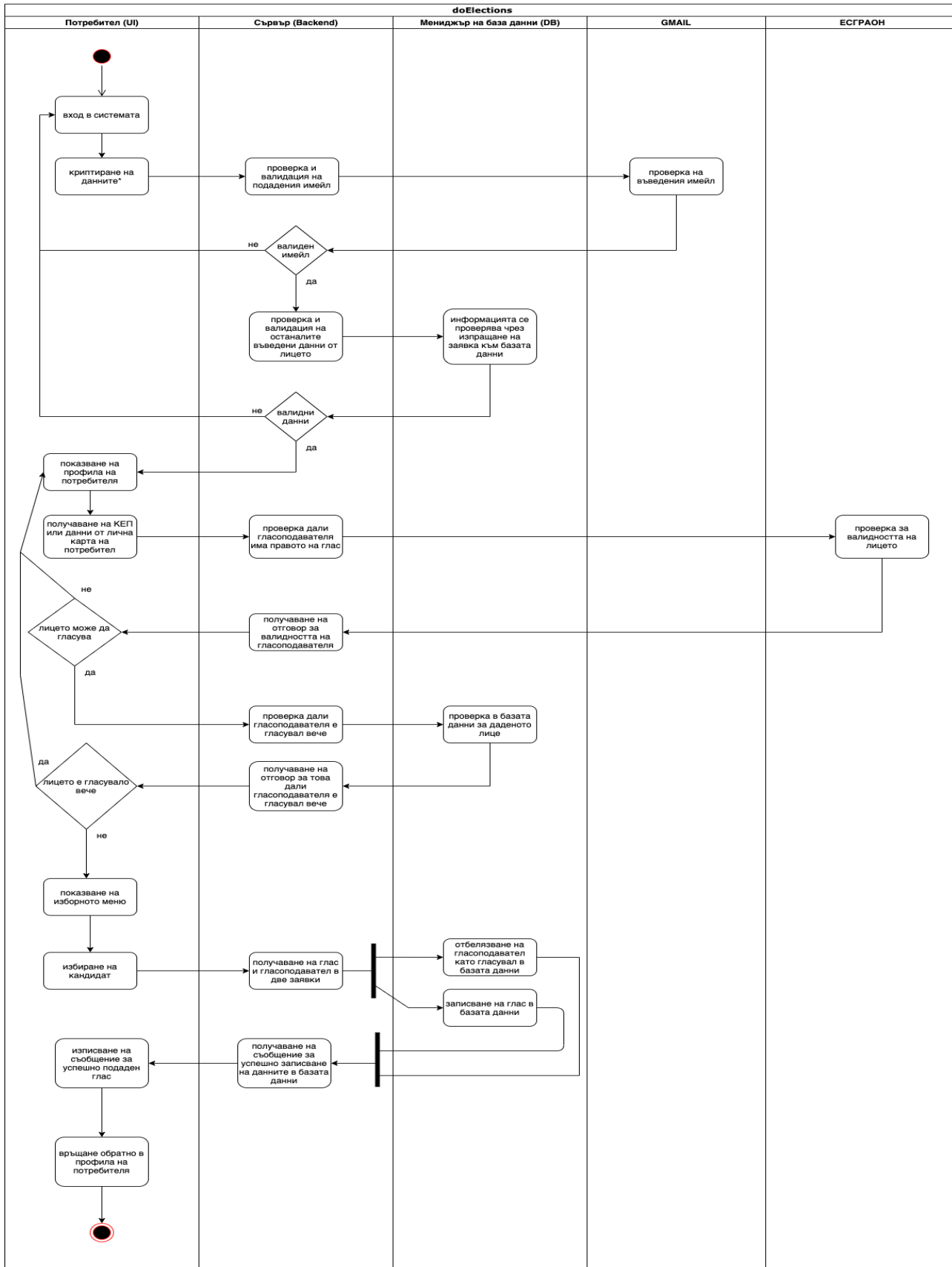
Осъществява се посредством „връзка с GMAIL“ от „Сървър“. Този подмодул от „комуникационен интерфейс“ е отговорен за изпращане на запитвания към GMAIL и получаване на отговорите от тях. Системата doElections прави запитвания към GMAIL в следните ситуации:

3.1.3.2.1 Проверява се дали въведеният от потребителя имейл адрес е валиден.

3.1.3.2.2 Изпраща се имейл на съответния потребител, в различни предварително определени ситуации (успешно гласуване, влизане в системата, запитване за възстановяване на парола, успешно възобновяване на парола и т.н.)

3.2. Структура на процесите

3.2.1. Първично представяне



*С цел да се опрости видът на диаграмата, криптирането на данните преди тяхното изпращане е представено единствено в началото. От тогава до края на диаграмата е пропуснат този процес, като се има в предвид че процесът на криптирането и изпращането на данните ще е един и същ всеки път, когато данните преминават от модул в друг модул.

Също така, мениджърът на базата данни в тази диаграма представлява интерфейса на базата данни. За да не се усложни диаграмата, мениджърът на базата данни репрезентира самата база данни. Единствените детайли, които са пропуснати, са заявките, които се изпращат към базата данни.

3.2.2. Описание на елементите и връзките

При влизането на (електронен) потребител в системата, той задължително трябва да се регистрира ако иска да гласува. В този случай, както е посочено при декомпозицията на модули, потребителя трябва да въведе имейл, парола, ЕГН и КЕП. В случая, когато в системата влезе регистриран гласоподавател, той трябва да въведе имейл и парола, а административното лице трябва да влезе като потребител, за да може да гласува. И в трите случая данните се валидират по следния начин: първоначално, данните се криптират и след това те се изпращат към сървъра за тяхната валидация. След това, сървърът изпраща заявка към GMAIL за проверка на валидността на въведения имейл. Ако той не е валиден, системата отказва опита за влизане на дадения потребител и се връща към логин/регистрация. В противен случай, се изпраща още една заявка към базата данни, за да се провери дали потребител със същите данни вече съществува в системата. Ако той съществува, системата отказва опита за влизане и се връща към логин/регистрация, а ако не - системата създава или влиза в профила на потребителя. От там той вече ще може да избере опцията гласуване (тя ще бъде отворена на изборния ден). Но преди да може да гласува, той трябва да се валидира, тоест ако е през приложението - той трябва да подаде своя КЕП, а ако той е от терминал - системата получава данни от скенера, които представляват информация от личната карта на гласоподавателя. И в двата случая данните се изпращат към сървъра, където те се обработват за валидност - дали гласоподавателят може да гласува. Това се проверява чрез заявка към ЕСГРАОН. Ако лицето бъде одобрено, то то се проверява отново, но този път чрез заявка към базата данни за определянето на това дали гласоподавателят е гласувал вече. В случаите, когато лицето няма правото на глас или той вече е гласувал, системата отхвърля достъпа на потребителя към гласуването и се връща към профила му. Иначе, потребителският интерфейс показва изборното меню. След като гласоподавателят избере някой от кандидатите и подаде своя глас, гласът се разбива на две заявки, които се изпращат една след друга. Едната съдържа информация за това кой е гласувал, а другата - за кого е било гласувано. И двете заявки се изпращат към сървъра, откъдето те се препращат към базата данни за съхранение. След това се изпраща съобщение към потребителския интерфейс от сървъра за успешно подаден глас и системата се връща към профила на потребителя.

3.2.3. Описание на обкръжението

На тази диаграма GMAIL и ЕСГРАОН са външни системи, както е описано в диаграмата за декомпозицията на модули. Те не са част от тази система, но към тях се изпращат заявки за валидации, които са част от процеса на гласуването:

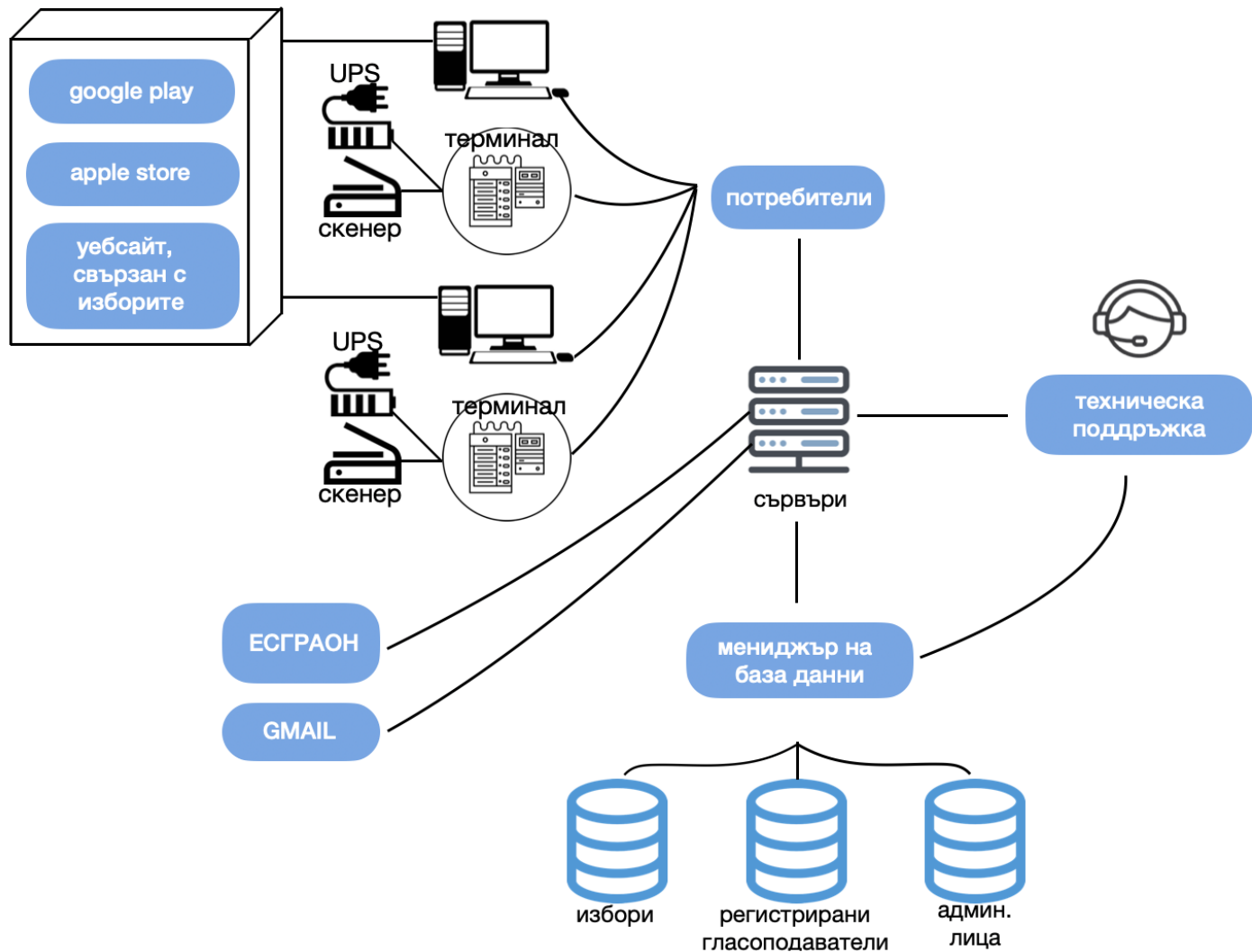
- При валидиране на гласоподавател за неговото право за гласуване, се изпраща заявка към ЕСГРАОН;
- При валидиране на въведен имейл от потребителя, системата се обръща към GMAIL.

3.2.4. Описание на възможните вариации

- Входът в системата може да се осъществи по два начина - или чрез логин, или чрез регистрация, при условие, че потребителя не притежава акаунт в системата. От друга страна, логин-а е разделен на две - за регистриран гласоподавател, където ще се иска единствено имейл и парола и за административно лице - ще се иска имейл, парола и КЕП;
- Представянето на профила на потребителя след неговото валидиране, силно зависи от типа на потребителя, който е влязъл в системата.
- В зависимост от това къде се гласува (от терминал или от приложението), системата получава или КЕП, или данни от личната карта на гласоподавателя и впоследствие се обработват различен тип данни.

3.3. Внедряване

3.3.1. Първично представяне



3.3.2. Описание на елементите и връзките

Приложението, чрез което потребителите ще получават достъп до системата, ще има само десктоп вариант. Това ще опрости системата и по този начин ще минимализира вратичките, от които ще може да се атакува. Предвид целите на системата и желаното високо ниво на сигурност, простотата може да внесе ограничение от към възможности за провал и атака.

Изискването за 100% наличност през целия изборен ден налага на модулите Сървъри и Мениджър на База Данни да са налични постоянно. За постигането на тази цел ще се наложи репликация на сървърите и презапасяване. Тъй като структурата ще бъде използвана твърде интензивно по време на изборен ден и почти неизползвана извън рамките му ще е необходимо и системата да бъде лесно скалируема за кратки периоди от време. За да постигнем разделение и сигурност в складирането на данни ще се наложи да използваме дистрибутирана база данни като на всяко от специално предназначенията за нея хардуерни устройства се разполага един от следните дялове: Регистрирани гласоподаватели, административни лица, избори.

Софтуерът, който ще позволява регистрация на потребителите, желаещи да дадат своя глас по електронен път от вкъщи, ще има възможност да се сваля от google play, apple store и също така ще има линкове за сваляне, разположени в сайта свързан с изборите. Например, ако изборите са за избиране на народни представители, сайта ще бъде този на министерството на вътрешните работи, ако пък изборите от друга страна са за конкурс за красота, то линковете ще са разположени в съответния сайт, свързан с конкурса.

В терминалите, които ще се намират в изборните секции ще има предварително инсталиран такъв софтуер, който ще позволява да се пускат заявки към сървъра без необходимост от регистрация както при онлайн регистрацията с персоналния компютър. Там обаче ще се налага да се премине през друг хардуерен компонент - скенер. Освен това, за гарантирането на 100% наличността на системата в изборния ден ще се прибегне до друг чисто хардуерен подход - използването на UPS (Uninterruptible Power Supply), който ще защитава данните и терминалното оборудване, като ни предпазва от прекъсване на електрозахранването или временното лошо качество на захранването до момента на отстраняването на проблема.

Имащи вече това браузърно приложение инсталирано успешно, типовете потребители изброени по-горе ще имат възможността да комуникират със сървърите на системата.

В „сървъри“ ще се намира сърцето на софтуера. Те ще са няколко, и ще могат да приемат заявки от най-близките рекуести до тях. Ще има няколко инстанции на сървърите, така че когато някой сървър спре да работи, другите сървъри да могат безпроблемно да поемат неговата част от работата, която е изпълнявал, като прием и обработка на заявките на потребителите. При преминаване на сървър към неактивен, техническият отдел ще има грижата да отстрани проблема и да върща в строя прекратилия работа сървър. Поддържането на активния им излишък и модула за следене на трафик и неизправност при техническата поддръжка ще гарантиран желаната наличност на системата. Библиотеката за криптиране ще осигурява сигурния пренос на данни.

Мениджърът на базата данни ще служи като интерфейс, хеширане и разпределяне на входните данни преди те да достигнат базата данни. За целта

базата данни ще е дистрибутирана измежду няколко хардуерни устройства, които ще гарантират липсата на свързаност между глас и гласоподавател, както и ще повишат сигурността. Съвървите на базата данни ще приемат директни заявки само от мениджъра на базата данни. Всяко физически отделно хардуерно устройство (съвър) ще отговаря за точно един дял от базата данни.

Модулът за техническата поддръжка ще бъде разположен на компютрите на екипа, който ще отговаря за това. За разлика от интерфейса, който предоставя модула на потребителя, който е thin client, този на техническата поддръжка ще е fat client и ще предоставя имплементирани функционалности. Това ще е под формата на специално за тях приложение. По този начин той ще може да си комуникира със съвървите и базата данни по специфичен в сравнение с потребителския начин и ще получава информация за трафика и изпълнението на съвървите, за да може да открие нередност в работата им. За него ще се осигурява достъп до количествени параметри по отношение на изпълнението на съвървите. Колко са активни в момента, какъв е трафика достигащ до тях, за какво време обработват данни от потребителите и извличат такива от външните системи.

3.3.3. Описание на обкръжението.

Външните системи представляват софтуер, който се отнася до нашата система, до такава степен, че тя не се грижи за физическото му разпределение. Използва се предоставената от техния интерфейс функционалност главно с цел валидация на използвани от нашата система данни.

С помощта на ЕСГРАОН ще валивираме личните данни като номер на лична карта и ЕГН както при регистриране в системата, така и по време на избори. Също така, когато дадено лице желае да гласува, се проверява дали то има това право.

Системата ще се допитва до GMAIL, за да се потвърди коректността на имейла на потребителя. Също така ще се предоставя възможност за изпращане на имейл в определени предварително дефинирани случаи - успешно гласуване, влизане в системата, запитване за възстановяване на парола, успешно възобновяване на парола и т.н.

Google play, Apple store и сайта, свързан с изборите, ще предоставят възможност да се инсталира нужният софтуер за връзка със системата doElections. В този софтуер ще бъде позициониран модула потребители, който ще е инсталиран и на терминалите в изборните секции.

3.3.4. Описание на възможните вариации.

Тъй като системата предполага голямо натоварване за малки периоди от време и много малко такава за останалата част от жизнения му цикъл (само представяне на статистики и други непредполагащи голяма активност услуги), то като вариация може да се добави допълнителен слой на автономност на системата и

параметрите, свързани с работата, трафика и натовареността на сървърите, до които техническия екип има достъп биха могли да се използват за автоматизирано скалиране на системата при необходимост. Това би могло да повиши производителността и да улесни работата на техническия екип.

4. Архитектурна обосновка

В тази секция ще защитим драйверите на системата, чрез архитектурните решения, които са взети.

4.1. Системата doElections трябва да поддържа няколко различни вида потребители.

4.1.1 Описание на драйвера

Един от най-важните аспекти на системата е да се разграничат различните видове потребители, всеки от които има различни права. Необходимо е да се различават потребителите, гласоподавателите, регистрираните гласоподаватели както и административните потребители (главен и регионален администратор, председател на секция). Основните функции на различните видове потребители са следните:

- потребител - може да гледа филтрирани статистики за избран ден;
- гласоподавател - може да гласува на място;
- регистриран гласоподавател - може да гласува чрез системата;
- административни потребители - осигуряват нормалното протичане на изборния ден чрез осигурените им от системата права.

4.1.2 Начин на осъществяване

В модула “Потребители” е обособен подмодул “профили”. Той изпълнява ролята на обвивен модул (wrapper), в който се разполагат всички видове лица, които ще използват системата. По този начин се постига разделение на различните функционалности, до които те ще имат достъп.

4.2. Всеки гласоподавател трябва да може да гласува по един от 3 начина

4.2.1. Описание на драйвера

Най-важната цел на системата doElections е осигуряването на 3 различни начина за гласуване на всеки гласоподавател. Ето защо е задължително извършването на всички необходими промени по архитектурата, за да се постигне това. Трябва да се предостави възможност на всеки, който има право на глас, да упражни това право по предпочитания от него начин:

- В избирателна секция посредством хартиена бюлетина, валидирайки си самоличността посредством лична карта.
- В избирателна секция посредством системата doElections, валидирайки си самоличността посредством лична карта.
- Дистанционно чрез системата doElections, валидирайки самоличността си посредством квалифициран електронен подпис (КЕП).

За да е възможно гласуването както с хартиена бюлетина, така и през терминал на място е необходимо системата doElections да осигурява връзка със специален скенер. Този скенер ще позволява удостоверяване на самоличността на лицето посредством неговата лична карта.

С цел осигуряване на възможност за дистанционно гласуване, системата doElections трябва да позволява регистриран гласоподавател да валидира самоличността си посредством квалифициран електронен подпис (КЕП).

4.2.2. Начин на осъществяване

Осигурена е разнородност за начина на подаване на данните, които се приемат от сървъра. Данните се приемат и в зависимост от вида си се правят съответните проверки. Изпраща се заявка към ЕСГРАОН, за да сме сигурни, че лицето има право на глас и към нашата база данни, за да знаем, че вече не е гласувало в рамките на текущите избори. Всичко това позволява използването на функционалността за гласуване по начина, по който всеки един от гласоподавателите желае.

4.3. Използване на данни от външни системи

4.3.1. Описание на драйвера

Осигуряването на начин за комуникация с външни системи (като ЕСГРАОН), от които ще се взима информацията за хората в дадена държава или област е жизненоважно за събирането на нужната информация за провеждането на изборите. Нейното своевременно актуализиране и проверка са от първостепенно значение за коректността на крайните резултати от гласуването.

4.3.2. Начин на осъществяване

Връзката с външните системи се осигурява чрез специализираните за това подмодули от “комуникационен интерфейс” в “Сървър”. По този начин се осъществява нужната информация за провеждането на коректни избори.

4.4. Системата да не обвързва по никакъв начин самоличността на даден гражданин с подадения от него глас

4.4.1. Описание на драйвера

Поради предназначението на doElections, от голямо значение е да се приоритизира сигурността на данните, които съхранява системата. Системата

трябва да запази анонимността на всеки гласоподавател - тя не трябва да обвързва гласоподавателя с подадения от него глас. Въпреки че това поставя ограничения върху начините на комуникация и в следствие съхраняване и обработване на информацията и следователно би повлияло на архитектурата, това изискване е от голямо значение за осигуряването на защитата на данните на системата doElections.

4.4.2. Начин на осъществяване

Взетото архитектурно решение е да се изпращат последователно 2 заявки от "Потребител" към "Сървър". Първата заявка ще бъде с информацията кой гласоподавател е дал своя вот. Втората заявка ще показва за кой от кандидатите е даденият вот. Тези две заявки се изпращат и обработват винаги в такава последователност, за да се гарантира, че не е възможно едно лице да гласува повече от веднъж в рамките на едни и същи избори.

4.5. Системата трябва да бъде 100% налична в рамките на даден изборен ден, дори при възможност за наличие на 100000 едновременни заявки.

4.5.1. Описание на драйвера

По време на един изборен ден, е от огромно значение в рамките на определения период от време системата doElections да бъде налична, за да може да се предостави шанс на всеки гласоподавател да гласува. Неотчитането на дори малък процент от вотовете може да бъде фатално и да промени резултатите от изборите, поради което това не трябва да се допуска. Практически, почти винаги, е невъзможно да се достигне 100% наличност на системата, поради което нашата цел ще е постигането на 99.99999% наличност от времето. Системата трябва да е проектирана по начин, който осигурява продължаване на нормалната ѝ работа дори в случаите, когато се появи грешка, предизвикана от отказ в хардуер, софтуер или комуникационно оборудване.

Архитектурата на системата трябва да позволява запазване на нормалната ѝ работа дори при по-високи нива на натоварване от очакваното. Тя трябва да издържа на натоварване от поне 100000 заявки в един и същ момент. В някои часове броят на гласуващите може да бъде много по-висок от очакваното. Въпреки това, системата трябва да осигури възможност на всеки гласоподавател да гласува като не се допуска срыв, предизвикан от множеството едновременни заявки.

4.5.2. Начин на осъществяване

При системата doElections ще се приложи комбинация от тактиките за репликация и активен излишък по отношение на сървърите. На тях се разчита за реализирането на необходимата функционалност на системата. По този начин, ако в даден сървър се появи проблем, един от останалите ще го замени, докато бива възстановяван или оправян от техническия екип. Наличието на няколко работещи сървъра едновременно, като всеки един от тях имат резерви, ще

предотврати възможността за downtime-а при възникване на грешка в един или няколко сървъра. Модулът „Наблюдаване“ осигурява нужната информация за трафика, натовареността и цялостното състояние на системата. Така ще се предостави възможност на всеки, желаещ да гласува, да упражни своето право.

4.6. Системата трябва да гарантира 100% интегритет на данните по време на изборния ден

4.6.1. Описание на драйвера

Системата ще съхранява вотовете на хора от гласуване, от което може да зависи бъдещето на дадена страна или организация. Ето защо, е от много голямо значение запазването на цялостта и достоверността на данните. Това означава, че не бива да се губят гласове или да се уврежда съдържанието им, тоест да показват грешна или неточна информация и по този начин да се извеждат резултати и статистики на базата на неточни вотове. Поради естеството на съхраняваната информация, всякакъв вид корупция на данните може да се окаже фатално. Затова doElections трябва да осигурява цялостност на данните минимум 99,99999%. Следователно, архитектурата на системата трябва да бъде изградена по начин, който осигурява надеждния поток от данни.

4.6.2. Начин на осъществяване

Базата данни поддържа лог на всички транзакции, които трябва да бъдат осъществени. По този начин дори при възникване на проблем при мениджъра на базата данни или временна недостъпност до базата данни или част от нея, се осигурява възможност за възстановяване и осъществяване на всички заявки, които е трябвало да бъдат извършени. Алгоритмите за криптиране, които се използват за осигуряването на сигурния пренос на данните, са ефективни, а вероятността за колизии е под $1/10^{18}\%$.

4.7. При генериране на заявка към системата без значение от натоварването, системата трябва да връща отговор в рамките на 1 секунда

4.7.1. Описание на драйвера

Архитектурата на doElections ще зависи от производителността, поради поставеното изискване за отговор в рамките на определено време. Това означава, че нашата система трябва да е изградена по такъв начин, че максималното време за отговор на всяка заявка да е не повече от 1 секунда.

4.7.2. Начин на осъществяване

В системата са предприети редица тактики, които да подобрят производителността и да осигурят постигането на необходимото бързодействие дори при пиково натоварване на системата. Във всеки един сървър, ще бъде възможно конкурентното изпълнение на множество заявки. Също така алгоритъмът за криптиране е ефективен, което предпазва системата от пагубни забавяния преди изпращане и след получаване на информацията в дадения

модул. Също така, поради наличието на няколко инстанции на сървъри, системата ще има възможност да се обработва заявките за максимум от 1 секунда, независимо от натовареността. За допълнително подобряване на производителността на системата е реализирана логика, гарантираща, че заявката на гласувачия се изпраща посредством “връзка със сървъра” до най-близкия до нея сървър.